An Analysis of Three Methods for Computing Weapon Scores and Importances PT. 1

by

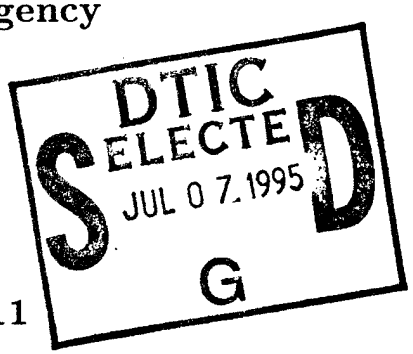
Dr. Peter F. Stiller
Professor of Mathematics and Computer Science and
Assistant Director of the
Institute for Scientific Computation
Texas A&M University

for the

United States Army Concepts Analysis Agency
8120 Woodmont Avenue
Bethesda, MD 20814-2797

and the

United States Army Research Office
4300 S. Miami Blvd.
P.O. Box 12211
Research Triangle Park, NC 27709-2211



under

Contract No. DAAH04-93-G-0310

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	30 March 1995	Technical 1 Aug 1993 - 31 Jan 1995

4. TITLE AND SUBTITLE

An Analysis of Three Methods for Computing Weapon Scores and Importances PT. 1

5. FUNDING NUMBERS

G: DAAH04-93-G-0310

6. AUTHOR(S)

Peter F. Stiller

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Texas A&M Research Foundation
Box 3578
College Station, TX 77843

8. PERFORMING ORGANIZATION REPORT NUMBER

8516 PT. 1

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Research Office
P. O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

ARO 31495.1-MA

11. SUPPLEMENTARY NOTES

The view, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This work analyzes three methods for computing scores for weapons in combat simulations. The eigenvalue or potential/antipotential method is based on a simultaneous eigenvalue problem. Fire allocation methods link fire allocation, values, and attrition in a non-linear way. The resulting systems of equations are surprising similar in behaviour to the linear ones that arise in the eigenvalue method. Techniques for solving these systems are discussed. Finally the weapon importance equations used in ATCAL (an attrition calibration procedure) are discussed. Existence and uniqueness of solutions is proved under some mild hypotheses. Again a surprising similarity with the linear methods is observed.

14. SUBJECT TERMS

eigenvalue method
potential/antipotential
fire allocation
Grobner bases
attrition
ATCAL
resultants

15. NUMBER OF PAGES
96

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

An Analysis of Three Methods for
Computing Weapon Scores and Importances

§0. Introduction.

The objective of this research is to provide new analytic approaches and new mathematical tools for combat modeling and simulation. In this paper we focus on three methods for assigning values (scores, importances) to various weapons in a simulated engagement. The potential/antipotential or eigenvalue method and the method used by the U.S. Army Concepts Analysis Agency as part of its ATCAL (Attrition Calibration) procedure, are part of current practice. The fire allocation methods are, relatively speaking, a new idea and lead to some very interesting systems of non-linear equations that can be viewed as non-linear generalizations of the simultaneous eigenvalue problem that occurs in the (linear) eigenvalue method. This similarity with the eigenvalue method is striking and leads one to conclude that fire allocation methods are the "right" non-linear generalization of the eigenvalue method.

We have been able to derive a number of useful results. For example, as part of our investigation of the ATCAL weapon importances, we were able to establish the existence and uniqueness of the solution to the importance equations. The proof, which is amazingly similar to the proof of the Perron-Frobenius Theorem that gives existence and uniqueness for the eigenvalue method, provides a new iterative scheme for solving the ATCAL importance equations. Again, the similarity with aspects of the (linear) eigenvalue method is interesting, and one is led to conclude that the ATCAL importances will inherit many of the same problems that occur with the eigenvalue method.

We also discuss various methods for dealing with the systems of non-linear equations that occur in the fire allocation methods. These problems take us to the frontiers of modern mathematical research into computational methods for finding resultants and Gröbner bases.

Finally, we conclude with some numerical experiments that tested the sensitivity of the three scoring methods to changes in various inputs and that made some comparisons between them. Because large examples would be computationally expensive, these experiments were run only for small examples having at most three weapons per side.

Clearly warfare involves many dynamically interacting "variables" and a large measure of unpredictable human behaviour. As a result, it is not all that suited to scien-

tific/mathematical analysis. Nevertheless, we feel that it is worthwhile to examine a variety of mathematical paradigms and techniques, so as to be able to build the richest possible models that can accommodate the widest range of phenomena. Moreover, the mathematical understanding of a model's behavior is critical to effectively moving results between models in a hierarchy. For those reasons, basic research into foundational issues in combat modeling and simulation is important.

The author would like to thank the U.S. Army Concepts Analysis Agency, particularly the Chief of its Advanced Research Projects Office (ARPO), Mr. Gerald E. Cooper, for their support of this research and their hospitality during visits to the agency.

§1. An Analysis of the Potential/Anti-Potential or Eigenvalue Method for Assigning Values (Scores) to Weapon Types.

a. The Method.

Numerous methods have been proposed for assigning "values" to weapons in combat models (Anderson [56]). Typically the resources of the two opposing sides are grouped into a number of generic types (e.g. M1A1, MLRS, BMP, TRUCK, etc.), with the object of the method being to compute a value (or score) for each weapon of a particular type. These values are then used to make comparisons across types and to make overall force comparisons.

Our purpose in considering here the Potential/Anti-Potential method and some of its variants is three-fold. First, we will use it to illustrate how some rather sophisticated mathematical ideas and important computational issues can arise in a relatively simple context. For example, Anderson [56] points out that some methods of assigning weapons scores are in everyday use, eventhough it is not known whether solutions of the equations used in that method always exist, and if they exist, whether or not they are unique (e.g. ATCAL, see Section 3 below). Second, it is always important to know the limitations of your mathematical methodology. Fortunately, with the Potential/Anti-Potential method we can describe those limitations precisely. In addition, whenever one selects a particular mathematical paradigm to use in building a model, one frequently can derive additional properties of the model as consequences of the initial assumptions. These derived features can serve as a kind of check on the reasonableness of the choice of mathematical technique/paradigm. Such "derived consequences" can be illustrated with the Potential/Anti-potential method. Our final purpose in considering this method is that many of its features carry over to the other two methods we will be discussing — fire allocation and ATCAL importances. Thus this method will serve to illustrate some of our later ideas. For all of these reasons, we have chosen to discuss the method in considerable detail.

We suppose that the two opposing forces (Blue and Red) are made up of a mix of weapons systems broken down into m types for the Blue force and n types for the Red force. Let B_i denote the number of weapons of type i in the Blue force ($i = 1, \ldots, m$) and let R_j denote the number of weapons of type j in the Red force ($j = 1, \ldots, n$). We denote

the sought after values by VB_i and VR_j. Thus $VB_i\,(VR_j)$ represents the value per type i (j) weapon in the Blue (Red) force. We also let K_{ij} denote the rate at which one Blue weapon of type i kills Red weapons of type j and let L_{ji} denote the rate at which one Red weapon of type j kills Blue weapons of type i. K_{ij} is therefore measured in units of Red weapons of type j per Blue weapon of type i per unit of time.

In a very general setting, the K_{ij} and L_{ji} would be functions of the B_i's and R_j's so that the values VB_i and VR_j would change as the composition of the Red and Blue forces changed. To the extent that the K_{ij} and L_{ji} are non-constant, the total value for each force

$$B = \sum_{i=1}^{m} VB_i \times B_i \quad \text{and} \quad R = \sum_{j=1}^{n} VR_j \times R_j$$

would be a non-linear function of the force components B_i and R_j and of time. However the eigenvalue method can be viewed as instantaneous so that K_{ij} and L_{ji} may be taken as constant.

The basic assumption of the potential/anti-potential or eigenvalue method is that the value per weapon of a particular type is proportional to the rate at which a weapon of that type destroys the value of the enemy's weapons. We can assume a single positive constant of proportionality C or separate positive constants CB and CR for the Blue and Red forces respectively. We also impose the obvious constraint that all the values be non-negative. This leads to the following system of $m + n$ equations:

$$CB \times VB_i = \sum_{j=1}^{n} K_{ij} VR_j \qquad i = 1, \ldots, m$$

$$CR \times VR_j = \sum_{i=1}^{m} L_{ji} VB_i \qquad j = 1, \ldots, n$$

or in matrix terms:

(1) $$CB \times VB = K \times VR$$

(2) $$CR \times VR = L \times VB$$

subject to the constraints $VB_i \geq 0$ for $i = 1, \ldots, m$, $VR_j \geq 0$ for $j = 1, \ldots, n$, $CR > 0$, and $CB > 0$. Note that all the K_{ij} and L_{ji} are always non-negative and that CB and

CR, in addition to VB and VR, are unknowns. Substituting (2) into (1) and vice versa yields the simultaneous eigenvalue problem

$$\lambda\, VB = M_{KL}\, VB \tag{3}$$

$$\lambda\, VR = M_{LK}\, VR \tag{4}$$

where $\lambda = CR \times CB$ is a scalar, M_{KL} is the $m \times m$ matrix product of K with L, and M_{LK} is the $n \times n$ matrix product of L with K. Note that both M_{KL} and M_{LK} are non-negative, that is, have non-negative entries. For future reference, we also note that this system (3), (4) is remarkably similar to a system of non-linear equations that occurs in the fire allocation method of Section 2.

Because M_{KL} and M_{LK} are obtained as the product of K and L in different orders, it can be shown that they have the same non-zero eigenvalues (counted with multiplicity). Moreover, the Perron-Frobenius Theorem (see below) implies that if M_{KL} and M_{LK} are irreducible, then our system (3) and (4) will have an essentially unique solution. Specifically, M_{KL} and M_{LK} will have a common positive real eigenvalue λ whose associated eigenvectors VB and VR will have all positive entries. This eigenvalue $\lambda = \lambda_{\max}$ will be the largest real eigenvalue of both M_{KL} and M_{LK}, and the corresponding eigenvectors VB and VR will be unique up to a positive scale factor. Simply put, there will be (up to scale) only one eigenvector for each of M_{KL} and M_{LK} with non-negative entries, and both will correspond to the largest eigenvalue. If either matrix is reducible, then zero values are possible, and the solution may no longer be unique.

It is important to note that *any* choice of positive eigenvectors VB and VR will provide a solution to our original system (1) and (2). To see this, suppose we have selected two such positive eigenvectors VB and VR, then

$$M_{KL} \times (K \times VR) = (K \times L) \times (K \times VR) = K \times M_{LK} VR = \lambda(K \times VR)$$

so that $K \times VR$ is also a non-zero non-negative eigenvector of M_{KL}. By the uniqueness result mentioned above, $K \times VR$ must therefore be a positive multiple, call it CB, of VB, so that

$$CB \times VB = K \times VR$$

as desired. A similar calculation gives

$$CR \times VR = L \times VB.$$

If instead, we select $\widetilde{VR} = r\ VR$, $r > 0$ and $\widetilde{VB} = b\ VB$, $b > 0$ as our eigenvectors, then

$$\widetilde{CB} \times \widetilde{VB} = K \times \widetilde{VR}$$

and

$$\widetilde{CR} \times \widetilde{VR} = L \times \widetilde{VB}$$

where $\widetilde{CB} = CB \cdot \frac{r}{b}$ and $\widetilde{CR} = CR \cdot \frac{b}{r}$, so that $\widetilde{CR} \cdot \widetilde{CB} = CR \cdot CB = \lambda_{\max}$ remains invariant. Notice that if $b = r$ then CR and CB do not change. We thus have two ways to view our two degrees of freedom: we can freely scale the eigenvectors VB and VR independently, or we can scale CB freely (and hence CA reciprocally) and then scale VB freely (in which event VR is uniquely determined).

Thus all we really have at this point is relative weapon values for Blue and Red $\left(\frac{VB_{i_1}}{VB_{i_2}} \text{ and } \frac{VR_{j_1}}{VR_{j_2}}\right)$ with no way to compare Blue weapons to Red weapons. One common normalization which overcomes this, is to set $CB = CR = \sqrt{\lambda_{\max}}$ (which in effect equates units of Red value with units of Blue value). One can then do relative comparisons across Red and Blue weapons types. Finally, to obtain fixed values, one can normalize one weapon type to have value = 1, which in effect amounts to a change of scale in the units of "value".

Whether we chose to normalize or not, it is easy to see that the quantity

$$\frac{\sqrt{CR} \times R}{\sqrt{CB} \times B},$$

where $R = \sum_{j=1}^{n} VR_j \times R_j$ and $B = \sum_{i=1}^{n} VB_i \times B_i$ are the total Red and Blue values respectively, is constant independent of our two scaling degrees of freedom. We shall see below that this quantity has a natural interpretation as the effective force ratio. It is a combination of the quantitative ratio $\frac{R}{B}$ and the qualitative ratio $\frac{CR}{CB}$. Note that doubling quantitative strength is equivalent to a four-fold increase in qualitative strength.

We remark that other normalization schemes have been criticized (Anderson [56]) on the grounds that dividing a weapon type into two "very similar" types should not upset

the force ratio. Those arguments appear to be based on interpreting $\frac{R}{B}$ as the force ratio, which is not the appropriate measure. Finally, note that redefining value to be $\sqrt{CR}\,VR_j$ and $\sqrt{CB}\,VB_i$ (i.e. $r = \sqrt{CR}$ and $b = \sqrt{CB}$) is equivalent to setting $\widetilde{CR} = \widetilde{CB} = \sqrt{\lambda_{\max}}$.

In general $CB(CR)$ involves units of $(\text{time})^{-1}$ and can be regarded as the rate at which one unit of Blue (Red) value kills units of Red (Blue) value. This is because

$$CB = \frac{\sum_j K_{ij} VR_j}{VB_i} \quad \text{for every} \quad i = 1, \ldots, m$$

so that

$$CB = \frac{\sum_i B_i \sum_j K_{ij} \times VR_j}{\sum_i B_i \times VB_i}.$$

The numerator is the units of Red value killed per unit time, and the denominator is total Blue value. Thus CB is the number of units of Red value killed per unit of Blue value per unit time. Hence, if CR and CB are different, the units of Red and Blue value are qualitatively different.

The invariant quantity $\sqrt{CB \times CR} = \sqrt{\lambda_{\max}} = I$ can be regarded as a measure of the intensity of combat. If we scale all kill rates K_{ij} and L_{ij} by c, then $K \times L$ and $L \times K$ scale by c^2, and the common maximum eigenvalue changes by a factor of c^2. This in turn means that I scales by c, which is appropriate for a measure of intensity. In addition, if we consider the Lanchester model for direct fire (the square law) using our interpretation of CB and CR as kill rates and $B(t)$ and $R(t)$ as units of Blue and Red value at time t, i.e.:

$$\begin{aligned} \frac{dR(t)}{dt} &= -CB \times B(t) \\ \frac{dB(t)}{dt} &= -CR \times R(t) \end{aligned} \tag{3}$$

then $\tau = \sqrt{CB \times CR}\,t$ is the unit of non-dimensional time (see Przemieniecki [40]). The quantity $\sqrt{CB \times CR}$ reflects the rate at which we trace the attrition path – a clear measure of intensity. ($\sqrt{CB \times CR}\,t$ appears as the argument in the cosh and sinh terms of the solutions of (3).) Finally the quantity $\Phi_0 = \frac{\sqrt{CR} \times R(0)}{\sqrt{CB} \times B(0)}$ is known as the superiority parameter (see Przemieniecki [40]). If $\Phi_0 > 1$, Red will annihilate Blue before being annihilated, and vice versa if $\Phi_0 < 1$. Thus Φ_0 is the initial effective force ratio.

We can also regard the potential/anti-potential method as a means of aggregation which is consistent with the Lanchester square law. Consider the Lanchester system of differential equations:

$$(4) \qquad \left(\frac{dB_1}{dt}, \cdots, \frac{dB_m}{dt}, \frac{dR_1}{dt}, \cdots, \frac{dR_n}{dt}\right) = -(B_1, \ldots, B_m, R_1, \ldots, R_n) \begin{pmatrix} 0 & K \\ L & 0 \end{pmatrix},$$

or

$$\begin{pmatrix} dB_1/dt \\ \vdots \\ dR_n/dt \end{pmatrix} = \begin{pmatrix} 0 & -L^T \\ -K^T & 0 \end{pmatrix} \begin{pmatrix} B_1 \\ \vdots \\ R_n \end{pmatrix}.$$

Here we have assumed that at all times t the kill rates are constant. If we fix CB and CR, then the values VB_i and VR_j will be constant up to some common positive scale factor. (Normalize the value of some weapon type, say VB_1, to 1 to remove this ambiguity, or carry it through the argument below.)

If one now defines the aggregate Blue value at time t by

$$B(t) = \sum_{i=1}^{m} VB_i \times B_i(t)$$

and the aggregate Red value at time t by

$$R(t) = \sum_{j=1}^{n} VR_j \times R_j(t),$$

then one can show that $B(t)$ and $R(t)$ satisfy:

$$(5) \qquad \begin{aligned} \frac{dB}{dt} &= -CR \times R \\ \frac{dR}{dt} &= -CB \times B. \end{aligned}$$

The proof of this fact is straight-forward (see Section 1c below).

This result says that given the eigenvalue method as a means of aggregation, (5) is a consistent aggregated attrition model (see Section 1c below). In effect, aggregation commutes with attrition – we can run the attrition model (4) for a period of time and aggregate the result, or first aggregate and then run the attrition model (5) – either way, the result will be the same!

Note that the converse is also true. If we seek a *linear* aggregation

$$B(t) = \sum_{i=1}^{m} VB_i \times B_i(t)$$

$$R(t) = \sum_{j=1}^{n} VR_j \times R_j(t)$$

where $B_i(t)$, $R_j(t)$ are the solutions to (4) and VB_i, VR_j are some constant "weights", then imposing the attrition model (5) on the aggregated forces compels the relations

$$CB \times VB_i = \sum_{j=1}^{n} K_{ij} VR_j$$

$$CR \times VR_j = \sum_{i=1}^{m} L_{ji} VB_i$$

i.e. the weights must be the ones derived from the eigenvalue method. (Again, see Section 1c below.)

We conclude this section with a number of additional observations:

1) If $L \times K$ and $K \times L$ are irreducible (as required by the Frobenius-Perron theorem) then one can show that

$$\frac{\partial \lambda_{\max}}{\partial K_{ij}} > 0 \quad \text{and} \quad \frac{\partial \lambda_{\max}}{\partial L_{ji}} > 0.$$

Thus increasing even one kill rate (holding the others constant) will increase $\lambda_{\max}$ and our measure of intensity $I = \sqrt{\lambda_{\max}}$. This is a derived result which lends credence to the eigenvalue method and our interpretation of I as intensity. For numerical computation of these derivatives see Section 4.

2) We have included in the last part of Section 1 a geometric proof of the Frobenius-Perron theorem that uses the Brouwer Fixed Point Theorem. This theorem says that every continuous map from the closed n-ball $B^n = \{(x_1, \ldots, x_n) \in \mathbf{R}^n \mid \sum_{i=1}^{n} x_i^2 \leq 1\}$ into itself has a fixed point. This leads to the consideration of more general fixed point techniques as a basis for assigning force values and/or aggregating force components. Fixed point methods involving contractive maps often lead to iterative procedures which permit the effective computation of the fixed point. This is the case with the

eigenvalue method, and as we shall see, it is also the case with the ATCAL weapon importances.

3) When the K_{ij} and L_{ji} depend on other variables, including possibly $B_1, \ldots, B_m$, $R_1, \ldots, R_n$, it is important to try to understand the resulting dependence of VB_i and VR_j on these variables. This is especially interesting in a stochastic setting where K_{ij}, L_{ji} are random variables. We can ask for the distribution of the VB_i, VR_j. We have investigated these questions for small examples and the results are presented in Section 4.

4) The potential/anti-potential method has come under various criticisms.

   a) First, it is clear that the values VB_i, VR_j are really measures of relative lethality and that vulnerability is not accounted for. That is because they are instantaneous measures. If the K_{ij}, L_{ji} depend on time, perhaps via a dependence on the current force structure B_i, R_j, then the values will vary over time. It is in this change over time where vulnerability enters.

   b) "... The numeric values of the scores are sometimes over sensitive to small changes in the input kill rate matrices. Zero score values sometimes occur for major weapons. Also, the method sometimes splits an engagement into two disconnected separated engagements ...".

   All of these problems stem from the failure of LK and KL to be irreducible (or from numerical problems that are the result of being close to reducible). Simple examples where such problems occur are given by:

$$K = \begin{pmatrix} K_{11} & 0 \\ 0 & K_{22} \end{pmatrix} \qquad L = \begin{pmatrix} L_{11} & 0 \\ 0 & L_{22} \end{pmatrix}$$

and

$$K = \begin{pmatrix} K_{11} & K_{12} \\ 0 & K_{22} \end{pmatrix} \qquad L = \begin{pmatrix} L_{11} & L_{12} \\ 0 & L_{22} \end{pmatrix}.$$

b. Relationships with Lanchester Attrition Models and Aggregation.

In previous reports, we have shown that aggregation cannot be considered in isolation from the attrition model. It is only by linking the two that we can account properly for the dynamics of force-on-force dependence in any theory of combat values. To illustrate

a case where compatibility can be achieved, consider the potential/anti-potential method explained above. We have

$$\begin{pmatrix} \frac{dB_1}{dt} \\ \vdots \\ \frac{dB_m}{dt} \\ \\ \frac{dR_1}{dt} \\ \vdots \\ \frac{dR_n}{dt} \end{pmatrix} = \begin{pmatrix} 0 & -L^T \\ \\ -K^T & 0 \end{pmatrix} \begin{pmatrix} B_1 \\ \vdots \\ B_m \\ R_1 \\ \vdots \\ R_n \end{pmatrix}$$

aggregating to

$$\frac{dB}{dt} = -CR \times R$$
$$\frac{dR}{dt} = -CB \times B$$

where $B = \sum_{i=1}^{m} VB_i \times B_i$ and $R = \sum_{j=1}^{n} VR_j \times R_j$ is the aggregation process.

If we assume that the K_{ij} and L_{ji} are constant, then the values VB_i and VR_j are constant and the Jacobian J of our aggregation process is the 2 by $m+n$ matrix

$$J = \begin{pmatrix} VB_1 & \dots & VB_n & 0 & \dots & 0 \\ 0 & \dots & 0 & VR_1 & \dots & VR_n \end{pmatrix}.$$

Now suppose our initial force composition is $(B_1^0, \dots, B_m^0, R_1^0, \dots, R_n^0)^T$ and that

$$\varphi(t) = (B_1(t), \dots, B_m(t), R_1(t), \dots, R_m(t))^T$$

is the "path" of attrition in the state-space $\mathbf{R}^{m+n}$. This means that $\varphi(t)$ satisfies

$$\frac{d\varphi}{dt} = \begin{pmatrix} 0 & -L^T \\ \\ -K^T & 0 \end{pmatrix} \varphi$$

and

$$\varphi(0) = (B_1^0, \dots, R_m^0)^T.$$

(These conditions uniquely determine φ.) Now consider the projection of this path

$$\gamma(t) = \left(\sum_{i=1}^{m} VB_i \times B_i(t), \quad \sum_{j=1}^{n} VR_j \times R_j(t) \right)^T$$

into the state-space $\mathbf{R}^2$. It satisfies

$$\frac{d\gamma}{dt} = \left(\sum_{i=1}^{m} VB_i \times \frac{dB_i}{dt}, \quad \sum_{j=1}^{n} VR_j \times \frac{dR_j}{dt} \right)^T .$$

(This is where the fact that K_{ij} and L_{ji} are constant comes into play.) In other words,

$$\frac{d\gamma}{dt} = J \cdot \frac{d\varphi}{dt}$$

which is the standard relationship between the velocity vectors of a path and those of its image under some map. On the other hand,

$$\frac{d\gamma}{dt} = \begin{pmatrix} 0 & -CR \\ -CB & 0 \end{pmatrix} \gamma$$

because of our choice of attrition model for the aggregated forces. The compatibility we have been discussing is the requirement that

$$\begin{pmatrix} 0 & -CR \\ -CB & 0 \end{pmatrix} \gamma = J \frac{d\varphi}{dt}.$$

For example, we must show

$$-CR \times R = \sum_{i=1}^{m} VB_i \cdot \frac{dB_i}{dt}.$$

But the right-hand side is $\frac{dB}{dt}$ – just differentiate the aggregation equation $B = \sum_{i=1}^{m} B_i \times VB_i$. Thus the projected path $\gamma(t)$ is the path of attrition in the aggregated variables B and R! This establishes consistency of the potential anti-potential method with the Lanchester attrition models.

c. The Frobenius-Perron Theorem.

In this part of Section 1, we discuss the Frobenius-Perron Theorem which plays a key role in the development of the so-called potential/anti-potential or eigenvalue method for assigning values (scores) to weapon types. It turns out that a very similar method, based on the same Brouwer fixed point theorem, can be used to establish existence and uniqueness

for ATCAL's weapon importances. We conclude our discussion with some observations on computational methods which implement the theorem.

Let A be a real $n \times n$ matrix with $a_{ij} \geq 0$ for every i, j:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}.$$

Such an A is said to be *non-negative.*

Definition 1. *A matrix A is said to be* reducible *if the index set $\{1, 2, \dots, n\}$ can be partitioned into two disjoint sets $\{i_1, \dots, i_\mu\}$ and $\{k_1, \dots, k_\nu\}$ $(\mu + \nu = n)$ such that*

$$a_{i_\alpha k_\beta} = 0 \quad \alpha = 1, \dots, \mu \quad \beta = 1, \dots, \nu.$$

In other words, after applying a suitable permutation to both the rows and columns of A, it takes the form

$$\begin{array}{c} \\ \mu \\ \\ \nu \end{array} \begin{array}{c} \begin{array}{ccc} \mu & & \nu \end{array} \\ \begin{pmatrix} * & \vdots & 0 \\ \dots & \dots & \dots \\ * & \vdots & * \end{pmatrix} \end{array}.$$

If a matrix is not reducible it is said to be irreducible. □

In 1907 Perron proved a remarkable theorem concerning the eigenvalues and eigenvectors of a positive matrix $(a_{ij} > 0)$. This was later generalized by Frobenius to the case of irreducible non-negative matrices.

Theorem 2. *(Perron-Frobenius.) An irreducible non-negative matrix $A = (a_{ij})$ has a unique largest real positive eigenvalue $\lambda_{\max}$ which is a simple root of the characteristic polynomial $x^n - tr A\, x^{n-1} + \dots + (-1)^n \det A$ of A. The modulus (complex absolute value) of any other eigenvalue will be less than or equal to $\lambda_{\max}$, and it will always be strictly less than $\lambda_{\max}$ if A is positive. To this maximal eigenvalue $\lambda_{\max}$ there corresponds a positive eigenvector $\vec{v} = (v_1, \dots, v_n)^T$ with $v_i > 0$ for every i. Of course, $\vec{v}$ is determined only up to a positive scalar.*

We will not prove this result in its full generality here. The reader should consult Gantmacher [20] for complete details. Instead, we will give a topological proof that makes use of Brouwer's fixed point theorem.

Let A be a real non-negative irreducible $n \times n$ matrix and let $\vec{x} = (x_1, \ldots, x_n)^T$, $x_i \geq 0$ all i, $\vec{x} \neq \vec{0}$ be a non-zero non-negative (column) vector. Since the theorem is trivial for $n = 1$, we will assume $n \geq 2$.

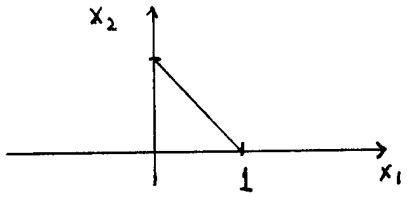
Lemma 3. $A\vec{x} \neq \vec{0}$.

Proof: $\vec{x}$ has at least one strictly positive component, say $x_j > 0$. If $A\vec{x}$ were to be $\vec{0}$, we would be forced to have $a_{ij} = 0$ for every $j = 1, \ldots, n$, but a column of zeros would make A reducible, contrary to our assumption. □

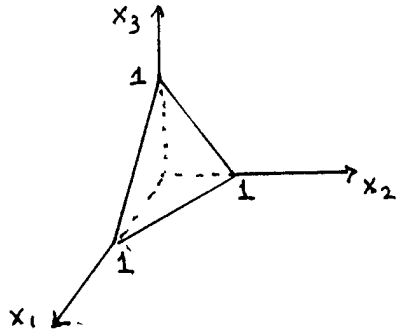
Recall that the standard $(n-1)$-simplex $\Delta^{n-1} \subset \mathbf{R}^n$ is defined by

$$\Delta^{n-1} = \left\{ (x_1, \ldots, x_n)^T \in \mathbf{R}^n \text{ s.t. } x_i \geq 0 \text{ for every } i \text{ and } \sum_{i=1}^{n} x_i = 1 \right\}.$$

For example, when $n = 2$, Δ^1 is a line segment in the plane:



and when $n = 3$, Δ^2 is a triangle in space:



Proof of Theorem 2. We define a continuous map g: $\Delta^{n-1} \to \Delta^{n-1}$ by

$$g(\vec{x}) = \frac{A\vec{x}}{\sigma(A\vec{x})}$$

where $\sigma((y_1,\ldots,y_n)^T) = \sum_{i=1}^{n} y_i$, i.e. σ means add the entries in the n-vector to which it is applied.

Note that σ: $\mathbf{R}^n \to \mathbf{R}$ is continuous, so that its reciprocal will be continuous away from places where it is zero. In fact, Lemma 3 shows that $\sigma(A\vec{x}) > 0$ for $\vec{x} \in \Delta^{n-1}$. Also note that $g(\vec{x})$ is again in Δ^{n-1} since $g(\vec{x})$ is a non-zero non-negative vector with

$$\sigma(g(\vec{x})) = \frac{\sigma(A\vec{x})}{\sigma(A\vec{x})} = 1.$$

Now Brouwer's Fixed Point Theorem says the following:

Let $B^n = \{\vec{x} \in \mathbf{R}^n \text{ s.t. } |\vec{x}|^2 = \sum_{i=1}^{n} x_i^2 \leq 1\}$ be the unit n-ball in $\mathbf{R}^n$ consisting of all vectors of length less than or equal to 1 in $\mathbf{R}^n$, and let f: $B^n \to B^n$ be any continuous map, then f has a fixed point, i.e. there exists a vector $\vec{x} \in B^n$ such that $f(\vec{x}) = \vec{x}$.

This theorem applies as well to any topological space which is topologically equivalent (homeomorphic) to B^n. This includes Δ^n. Thus our map g: $\Delta^{n-1} \to \Delta^{n-1}$ has a fixed point $\vec{x} \in \Delta^{n-1}$. We have

$$\frac{A\vec{x}}{\sigma(A\vec{x})} = \vec{x},$$

or, setting $\lambda = \sigma(A\vec{x}) > 0$,

$$A\vec{x} = \lambda\vec{x}.$$

In other words $\lambda > 0$ is an eigenvalue for A with non-negative eigenvector $\vec{x}$. In fact, every component of $\vec{x}$ must be strictly positive. To see this, suppose to the contrary that $\vec{x}$ has some 0 entries. Without loss of generality, we can permute the coordinate axes so that $\vec{x} = (x_1,\ldots,x_k,0,\ldots,0)^T$ with $x_i \neq 0$ for $i = 1,\ldots,k$ where $1 \leq k < n$. We then have

$$A\vec{x} = \begin{pmatrix} A_{11} & \vdots & A_{12} \\ \cdots & \cdots & \cdots \\ A_{21} & \vdots & A_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_k \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} \lambda x_1 \\ \vdots \\ \lambda x_k \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

so that $A_{21}\begin{pmatrix} x_1 \\ \vdots \\ x_k \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$. However, this relation, plus the fact that $x_1 > 0$,

$x_2 > 0, \ldots, x_k > 0$, forces A_{21} to be an $(n-k) \times k$ block of zeros, making A a reducible matrix.

We have thus established that a real non-zero non-negative irreducible matrix A has a positive eigenvalue with corresponding positive eigenvector. Additional arguments are needed to show that this eigenvalue has the largest absolute value among the real eigenvalues of A, and that no other real eigenvalue has a positive eigenvector. Notice that the fixed points of g are in one-to-one correspondence with the "non-negative" eigenvectors $\vec{y} = (y_1, \ldots, y_n)^T$, $y_i \geq 0$ all i, of A (up to positive scale), and that the argument above shows that all the fixed points lie in the interior of Δ^{n-1}, which is equivalent to $y_i > 0$ for all i. The theorem of course implies that g has a unique fixed point.

We remark that reducible non-negative matrices will still have non-negative eigenvalues with non-negative eigenvectors, however there may be several such. Examples to think about are:

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}.$$

Computational Issues

One can compute the unique fixed point of the map g: $\Delta^{n-1} \to \Delta^{n-1}$, where $g(\vec{x}) = \frac{A\vec{x}}{\sigma(A\vec{x})}$, by an iterative procedure. For this to work, we must assume that A, in addition to being non-negative and irreducible, is primitive. If all the entries of A are positive, then A is primitive. If A has some zero entries, then A will be primitive if and only if some power of A is positive.

To compute the fixed point, we pick any initial vector $\vec{x}^{(1)} \in \Delta^{n-1}$, say $\vec{x}^{(1)} = (x_1^{(1)}, \ldots, x_n^{(1)})^T$ where $x_j^{(1)} \geq 0$ all $j = 1, \ldots, n$ and $\sum_{j=1}^{n} x_j^{(1)} = 1$, and set

$$\vec{x}^{(i+1)} = g(\vec{x}^{(i)})$$

so that

$$\vec{x}^{(2)} = g(\vec{x}^{(1)}) \quad \vec{x}^{(3)} = g(\vec{x}^{(2)}) = g(g(\vec{x}^{(1)}))$$

and so on. One can show that:

Proposition 4. $\lim_{i\to\infty} \vec{x}^{(i)} = \vec{x}_0$ *where* $g(\vec{x}_0) = \vec{x}_0$ *is the unique fixed point of* g *and therefore, up to scale, the unique positive eigenvector of* A.

This iterative method breaks down if A is reducible or if A is not primitive, as the example:

$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

shows.

Sensitivity

Let A be a non-negative irreducible matrix. The largest eigenvalue $\lambda_{\max}$ can be characterized as follows:

$$\lambda_{\max} = \max_{\vec{x}\in\Delta^{n-1}} \left(\min_{\substack{1\le i\le n \\ x_i\neq 0}} \frac{(A\vec{x})_i}{x_i} \right)$$

where the maximum is taken over all vectors $\vec{x} = (x_1, \ldots, x_n)^T$ with $x_i \geq 0$ for all i and $\sum_{i=1}^{n} x_i = 1$ and where $(A\vec{x})_i = \sum_{j=1}^{n} a_{ij}x_j$ is the i^{th} component of the vector $A\vec{x}$ (see Gantmacher [ ]). Moreover, the unique vector $\vec{x}^{(0)}$ in Δ^{n-1} giving the maximum is in the interior of Δ^{n-1} and is an eigenvector with eigenvalue $\lambda_{\max}$:

$$A\vec{x}^{(0)} = \lambda_{\max}\vec{x}^{(0)}.$$

Now suppose every element in the j^{th} column of A is increased by $\varepsilon > 0$ yielding a new non-negative irreducible matrix $A_{j,\varepsilon}$. We compute that $(A_{j,\varepsilon}\vec{x}^{(0)})_i = (A\vec{x}^{(0)})_i + \varepsilon x_j^{(0)}$ so that

$$\min_{1\le i\le n} \frac{(A_{j,\varepsilon}\vec{x}^{(0)})_i}{\vec{x}_i^{(0)}} \geq \lambda_{\max} + \min_{1\le i\le n} \varepsilon \frac{x_j^{(0)}}{x_i^{(0)}}.$$

It follows that $\tilde{\lambda}_{\max}$, the largest eigenvalue for $A_{j,\varepsilon}$ is at least $\lambda_{\max} + \varepsilon \min_{1\le i\le n} \frac{x_j^{(0)}}{x_i^{(0)}} \gneq \lambda_{\max}$.

Now suppose in addition that A is primitive. From the definition of primitive, it follows that $A^q = (a_{ij}^{(q)})$ has only positive entries for some $q \geq 1$. If we increase the i,j-entry of A by ε, then every entry in the j^{th} column of $A^{q+1} = A^q A$ increases by at least $(\min_{1\le k\le n} a_{ki}^{(q)})\varepsilon$. The argument above then shows that the largest eigenvalue of A^{q+1} (which

is $\lambda_{\max}^{q+1}$) increases by at least $c_0\varepsilon$ where c_0 is fixed positive constant independent of ε. It follows that

$$c_0 \leq \frac{\partial \lambda_{\max}^{q+1}}{\partial a_{ij}} = (q+1)\lambda_{\max}^{q} \frac{\partial \lambda_{\max}}{\partial a_{ij}}$$

so that $\frac{\partial \lambda_{\max}}{\partial a_{ij}}$ is positive. Moreover, we can give a positive lower bound for this value. Numerical confirmation of this result can be found in Section 4.

§2. Fire Allocation Methods for Assigning Weapon Values

a. The Methods.

i. Introduction to Fire Allocation Methods.

Fire allocation methods for assigning values to the weapons present in an engagement are based on the principle that the value of opposing weapons at an instant in time will determine (along with other factors/variables) how we allocate the fire of friendly weapons. This allocation of fire simultaneously impacts the rate of attrition of various weapon types, so we will need attrition models that capture this effect. Finally attrition rates (at a given instant in time) effect value at that instant, because we will assume, as we did in the eigenvalue method (see §1), that the value of a weapon is proportional to the rate at which it is destroying enemy value.

As one might expect, working with models and equations that portray such a complex interaction between value, fire allocation, and attrition can be difficult. Unlike the relatively simple linear equations and matrix techniques of the eigenvalue method, fire allocation methods invariably lead to non-linear equations with parameters. In this section, we will illustrate fire allocation methods with examples that use a Lanchester-like attrition model. Other attrition models could be used. In fact, in a very general sense, the ATCAL model (see §3) also embodies a fire allocation method.

Because the examples lead to complex systems of non-linear polynomial equations, we also spend some time in this section discussing methods to solve these equations. For relatively few weapon types on each side, such systems of non-linear equations can be solved with the aid of a mathematical computation/symbolic computation package such as Mathematica or Maple. Finally, we can observe that despite the presence of non-linear equations in our fire allocation examples, the method exhibits features that are surprising similar to the eigenvalue method.

ii. A Simple Model.

We first consider a simple example with two weapon types on each side. We denote by $b_1(t)$, $b_2(t)$ the number of blue weapons of type 1 and type 2 respectively, present in the engagement at time t. Likewise $r_1(t)$, $r_2(t)$ are the respective numbers of each type of

red weapon present at time t.

Our attrition model will be a simple Lanchester square law based on the ordinary differential equations:

$$\frac{db_1}{dt} = -p_1(c_{11}r_1 + c_{12}r_2)$$
$$\frac{db_2}{dt} = -p_2(c_{21}r_1 + c_{22}r_2)$$
$$\frac{dr_1}{dt} = -q_1(a_{11}b_1 + a_{12}b_2)$$
$$\frac{dr_2}{dt} = -q_2(a_{21}b_1 + a_{22}b_2).$$

The c_{ij} and a_{ij} are attrition rates that reflect the rate at which one unit of a particular weapon destroys units of another. The p_k and q_ℓ represent an allocation of fire. Specifically, we will have $p_1, p_2,\ q_1, q_2 \geq 0$ with $p_1 + p_2 = 1$ and $q_1 + q_2 = 1$.

In this particular model we are assuming both red weapon types fire at a given blue target in the same proportion and likewise for blue. A more realistic model would perhaps be

$$\frac{db_1}{dt} = -c_{11}p_{11}r_1 - c_{12}p_{12}r_2$$
$$\frac{db_2}{dt} = -c_{21}p_{21}r_1 - c_{22}p_{22}r_2$$
$$\frac{dr_1}{dt} = -a_{11}q_{11}b_1 - a_{12}q_{12}b_2$$
$$\frac{dr_2}{dt} = -a_{21}q_{21}b_1 - a_{22}q_{22}b_2$$

where $p_{k\ell} \geq 0, q_{k\ell} \geq 0$ with $p_{1k} + p_{2k} = 1$ for $k = 1, 2$ and $q_{1\ell} + q_{2\ell} = 1$ for $\ell = 1, 2$.

In general p_1, p_2, q_1, q_2 will be functions of various parameters. To illustrate the method, we select an allocation of fire that assumes we fire at each target in proportion to its value. Thus if a unit of b_1 is twice as valuable as a unit of b_2, then 2/3 of red's fire will be directed at type b_1 and 1/3 at type b_2. Specifically, we set

$$p_1 = \frac{h_1 b_1(0)}{h_1 b_1(0) + h_2 b_2(0)}$$
$$p_2 = \frac{h_2 b_2(0)}{h_1 b_1(0) + h_2 b_2(0)}$$
$$q_1 = \frac{g_1 r_1(0)}{g_1 r_1(0) + g_2 r_2(0)}$$
$$q_2 = \frac{g_2 r_2(0)}{g_1 r_1(0) + g_2 r_2(0)}$$

where h_1 is the "value" of one unit of b_1 at the current time (which we take as $t = 0$ for simplicity) and h_2 is the "value" of one unit of b_2. Thus $h_1 b_1(0) + h_2 b_2(0)$ is the total "value" of blue at the current time. Similarly g_1, g_2 are the values for red at the current time.

This may or may not be a "realistic" model of fire allocation. What is important is that the fire allocation factors (p_k and q_ℓ in this case are functions of the unknown values and other known quantities ($b_i(0)$ and $r_j(0)$ in this case). For a discussion of the general case, see Section 2.a.v below.

As with the eigenvalue method, we assume that value is proportional to the rate at which a weapon destroys opposing value. For example,

$$\lambda h_1 = g_1 \left(\frac{g_1 r_1(0)}{g_1 r_1(0) + g_2 r_2(0)} \right) a_{11} + g_2 \left(\frac{g_2 r_2(0)}{g_1 r_1(0) + g_2 r_2(0)} \right) a_{21}$$

which says that the value of one unit of b_1 is proportional to the rate at which it destroys red value. Note that the units on the constant of proportionality λ are $(\text{time})^{-1}$. This equation can be viewed as being derived from

$$\lambda h_1 b_1 = g_1 \left(\frac{g_1 r_1(0)}{g_1 r_1(0) + g_2 r_2(0)} \right) a_{11} b_1 + g_2 \left(\frac{g_2 r_2(0)}{g_1 r_1(0) - g_2 r_2(0)} \right) a_{21} b_1$$

which says that the total value of all weapons of type b_1 is proportional to the rate at which they collectively destroy red value at the current time.

This leads to the following system of non-linear equations:

$$\lambda h_1 = \frac{a_{11} g_1^2 r_1(0) + a_{21} g_2^2 r_2(0)}{g_1 r_1(0) + g_2 r_2(0)}$$

$$\lambda h_2 = \frac{a_{12} g_1^2 r_1(0) + a_{22} g_2^2 r_2(0)}{g_1 r_1(0) + g_2 r_2(0)}$$

$$\lambda g_1 = \frac{c_{11} h_1^2 b_1(0) + c_{21} h_2^2 b_2(0)}{h_1 b_1(0) + h_2 b_2(0)}$$

$$\lambda g_2 = \frac{c_{12} h_1^2 b_1(0) + c_{22} h_2^2 b_2(0)}{h_1 b_1(0) + h_2 b_2(0)}$$

or

$$\lambda r_1(0) h_1 g_1 + \lambda r_2(0) h_1 g_2 - a_{11} r_1(0) g_1^2 - a_{21} r_2(0) g_2^2 = 0$$

$$\lambda r_1(0) h_2 g_1 + \lambda r_2(0) h_2 g_2 - a_{12} r_1(0) g_1^2 - a_{22} r_2(0) g_2^2 = 0$$

$$\lambda b_1(0) h_1 g_1 + \lambda b_2(0) h_2 g_1 - c_{11} b_1(0) h_1^2 - c_{21} b_2(0) h_2^2 = 0$$

$$\lambda b_1(0) h_1 g_2 + \lambda b_2(0) h_2 g_2 - c_{12} b_1(0) h_1^2 - c_{22} b_2(0) h_2^2 = 0.$$

We see that for each λ, we have a system of four homogeneous quadratic equations in four variables h_1, h_2, g_1, g_2. In general we expect such a system to have no solutions other than the trivial one $h_1 = 0$, $h_2 = 0$, $g_1 = 0$, $g_2 = 0$. However for certain values of λ (the analogue of an eigenvalue in the linear case) there will be non-trivial solutions. The problem is to determine what values of $\lambda > 0$ cause the associated system to have a solution with $h_1 \geq 0$, $h_2 \geq 0$, $g_1 \geq 0$, $g_2 \geq 0$. Thus our fire allocation method is a non-linear analogue of the eigenvalue method. Note that value is determined only up to scale, so that we must normalize the value relative to one type (which then has value 1). There are two procedures for solving this problem. The first treats the whole system of equations and uses classical elimination theory to compute a polynomial $R(\lambda)$ whose roots are the values of λ for which the system has a non-trivial solution. The coefficients of $R(\lambda)$ are polynomial expressions in a_{ij}, c_{ji}, $b_i(0)$ and $r_j(0)$. The specific computational device for finding $R(\lambda)$ is known as the Macaulay resultant and it is discussed below.

A second approach leads to two sets of equations analogous to a simultaneous eigenvalue problem

$$P_1(\lambda, h_1, h_2) = 0$$
$$P_2(\lambda, h_1, h_2) = 0$$
$$Q_1(\lambda, g_1, g_2) = 0$$
$$Q_2(\lambda, g_1, g_2) = 0$$

where the first set of equations $P_1 = 0$, $P_2 = 0$ involves only blue values and λ and the second set Q_1, Q_2 involves only red values and λ. These equations are quartic, homogeneous in h_1, h_2 or g_1, g_2 and are derived by substitution. For example, using

$$h_1 = \frac{a_{11} r_1(0) g_1^2 + a_{21} r_2(0) g_2^2}{\lambda_1 r_1(0) g_1 + \lambda r_2(0) g_2}$$
$$h_2 = \frac{a_{12} r_1(0) g_1^2 + a_{22} r_2(0) g_2^2}{\lambda r_1(0) g_1 + \lambda r_2(0) g_2}$$

and substituting into the equations for g_1 and g_2 yields:

$$\lambda g_1 = \frac{c_{11}b_1(0)\left(\frac{a_{11}r_1(0)g_1^2+a_{21}r_2(0)g_2^2}{\lambda r_1(0)g_1+\lambda r_2(0)g_2}\right)^2 + c_{21}b_2(0)\left(\frac{a_{12}r_1(0)g_1^2+a_{22}r_2(0)g_2^2}{\lambda r_1(0)g_1+\lambda r_2(0)g_2}\right)^2}{b_1(0)\left(\frac{a_{11}r_1(0)g_1^2+a_{21}r_2(0)g_2^2}{\lambda r_1(0)g_1+\lambda r_2(0)g_2}\right) + b_2(0)\left(\frac{a_{12}r_1(0)g_1^2+g_{22}r_2(0)g_2^2}{\lambda r_1(0)g_1+\lambda r_2(0)g_2}\right)}$$

$$\lambda g_2 = \frac{c_{12}b_1(0)\left(\frac{a_{11}r_1(0)g_1^2+a_{21}r_2(0)g_2^2}{\lambda r_1(0)g_1+\lambda r_2(0)g_2}\right)^2 + c_{22}b_2(0)\left(\frac{a_{12}r_1(0)g_1^2+a_{22}r_2(0)g_2^2}{\lambda r_1(0)g_1+\lambda r_2(0)g_2}\right)^2}{b_1(0)\left(\frac{a_{11}r_1(0)g_1^2+a_{21}r_2(0)g_2^2}{\lambda r_1(0)g_1+\lambda r_2(0)g_2}\right) + b_2(0)\left(\frac{a_{12}r_1(0)g_1^2+g_{22}r_2(0)g_2^2}{\lambda r_1(0)g_1+\lambda r_2(0)g_2}\right)}.$$

Clearing denominators gives

$$\begin{aligned}
0 &= \lambda^2 g_1(r_1(0)g_1 + r_2(0)g_2)\left[(a_{11}b_1(0)r_1(0) + a_{12}b_2(0)r_1(0))g_1^2\right. \\
&\quad \left.+(a_{21}b_1(0)r_2(0) + a_{22}b_2(0)r_2(0))g_2^2\right] \\
&\quad - c_{11}b_1(0)(a_{11}r_1(0)g_1^2 + a_{21}r_2(0)g_2^2)^2 - c_{21}b_2(0)(a_{12}r_1(0)g_1^2 + a_{22}r_2(0)g_2^2)^2 \\
0 &= \lambda^2 g_2(r_1(0)g_1 + r_2(0)g_2)\left[(a_{11}b_1(0)r_1(0) + a_{12}b_2(0)r_1(0))g_1^2\right. \\
&\quad \left.+(a_{21}b_1(0)r_2(0) + a_{22}b_2(0)r_2(0))g_2^2\right] \\
&\quad - c_{12}b_1(0)(a_{11}r_1(0)g_1^2 + a_{21}r_2(0)g_2^2)^2 - c_{22}b_2(0)(a_{12}r_1(0)g_1^2 + a_{22}r_2(0)g_2^2)^2.
\end{aligned}$$

The right had sides are $Q_1(\lambda, g_1, g_2)$ and $Q_2(\lambda, g_1, g_2)$ respectively.

Again, the system

$$P_1(\lambda, h_1, h_2)$$

$$P_2(\lambda, h_1, h_2)$$

generally has no solutions. However for certain values of λ it will. Likewise for

$$Q_1(\lambda, g_1, g_2) = 0$$

$$Q_2(\lambda, g_1, g_2) = 0.$$

In each case there is a polynomial in λ, $R_P(\lambda)$ or $R_Q(\lambda)$, whose roots are those λ for which that particular system has a non-trivial solution. Optimally, we will find a unique positive λ, call it $\lambda_{\max}$, with

$$R_P(\lambda_{\max}) = 0 \quad \textbf{and} \quad R_Q(\lambda_{\max}) = 0$$

which will lead to a unique solution for the weapon values (after normalizing). We cannot as yet prove this.

iii. An Iterative Scheme with Shifting Fires and Changing Values.

Before giving a specific numerical example and discussing more general fire allocation schemes for determining weapon values, we note that the simple method above leads to an interesting iterative time-step scheme that links value, attrition, and fire allocation.

At time t_0 with forces $b_1(t_0)$, $b_2(t_0)$, $r_1(t_0)$, $r_2(t_0)$ present in the engagement, we calculate the current values $h_1(t_0)$, $h_2(t_0)$, $g_1(t_0)$, $g_2(t_0)$. With these values known we can update the attrition for a time step Δt by setting

$$b_1(t_0 + \Delta t) = b_1(t_0) + \Delta b_1$$

etc., where the attrition over the time step is:

$$\Delta b_1 = -\left(\frac{h_1(t_0)b_1(t_0)}{h_1(t_0)b_1(t_0) + h_2(t_0)b_2(t_0)}\right)(a_{11}r_1(t_0) + a_{12}r_2(t_0))\Delta t$$

etc. This updates the forces present at time $t_1 = t_0 + \Delta t$. We now cycle through the process again. With this sort of model, we expect to see values change and fires reallocate as losses occur.

To obtain a time independent value for a unit of say b_1, we could average its value over time

$$\tilde{h}_1 = \frac{1}{t_{\text{final}} - t_{\text{init}}} \int_{t_{\text{init}}}^{t_{\text{final}}} h_1(t)dt$$

or

$$\frac{1}{n\Delta t}\left(\frac{1}{2}h_1(t_{\text{init}}) + h_1(t_{\text{init}} + \Delta t) + \cdots + h_1(t_{\text{init}} + (n-1)\Delta t) + \frac{1}{2}h_1(t_{\text{init}} + n\Delta t)\right)$$

where $n\Delta t = t_{\text{final}} - t_{\text{init}}$ is the duration of the engagement which begins at time t_{init} and ends at time t_{final}. Alternatively, we could compute a somewhat different average

$$\tilde{\tilde{h}}_1 = \frac{1}{b_1(0)(t_{\text{final}} - t_{\text{init}})} \int_{t_{\text{init}}}^{t_{\text{final}}} b_1(t)h_1(t)dt$$

which measures the average contribution to blue's total value by a single unit of b_1 at the start of the engagement over the duration of the engagement. Weapons with high vulnerability will not fare as well as weapons of equal lethality but less vulnerability with this measure of value.

iv. A Numerical Example of the Procedure for Solving the Simultaneous System of Non-linear Equations Arising in our Simple Fire Allocation Model.

Let's take our initial force strength to be

$$b_1(0) = 100 \qquad r_1(0) = 150$$

$$b_2(0) = 40 \qquad r_2(0) = 50$$

with attrition coefficients

$$a_{11} = .05 \qquad a_{12} = .03$$

$$a_{21} = .02 \qquad a_{22} = .07$$

$$c_{11} = .06 \qquad c_{12} = .04$$

$$c_{21} = .03 \qquad c_{22} = .08$$

so that the attrition model becomes

$$\frac{db_1}{dt} = -\left(\frac{100h_1}{100h_1 + 40h_2}\right)(.06r_1 + .04r_2)$$

$$\frac{db_2}{dt} = -\left(\frac{40h_2}{100h_1 + 40h_2}\right)(.03r_1 + 08.r_2)$$

$$\frac{dr_1}{dt} = -\left(\frac{150g_1}{150g_1 + 50g_2}\right)(.05b_1 + .03b_2)$$

$$\frac{dr_2}{dt} = -\left(\frac{50g_2}{150g_1 + 50g_2}\right)(.02b_1 + .07b_2).$$

Note that the red weapons have the better kill rates and can be expected to have better values. Thus g_1, g_2 should be larger than h_1, h_2 respectively.

Our equations for the values are

1) $$\lambda h_1 = \frac{7.5g_1^2 + 1.0g_2^2}{150g_1 + 50g_2}$$

2) $$\lambda h_2 = \frac{4.5g_1^2 + 3.5g_2^2}{150g_1 + 50g_2}$$

3) $$\lambda g_1 = \frac{6.0h_1^2 + 1.2h_2^2}{100h_1 + 40h_2}$$

4) $$\lambda g_2 = \frac{4.0h_1^2 + 3.2h_2^2}{100h_1 + 40h_2}.$$

After clearing denominators, we are led to a system of four homogeneous quadratic equations in four variables h_1, h_2, g_1, g_2. We treat λ as a coefficient rather than a variable. The

Macaulay resultant of this system will be a polynomial in λ whose roots are those values of λ for which the system has a solution. (See Section 2b for details on the Macaulay resultant.)

An alternative approach that is computationally more tractable, is to solve equations 1) and 2) for h_1 and h_2 and substitute those expressions into 3) and 4). Likewise we can solve 3) and 4) for g_1 and g_2 and substitute into 1) and 2). This leads, after some simple algebra, to two systems of two equations each linked by λ. Specifically, if we let $u = h_1/h_2$ and $v = g_1/g_2$, we arrive at

$$5) \qquad 0 = (-286 + 1000\lambda^2)u^4 + (44000\lambda^2)u^3 + (-133.6 + 34000\lambda^2)u^2 + (13600\lambda^2)u + (-21.04)$$

$$6) \qquad 0 = (-218)u^4 + (110000\lambda^2)u^3 + (-154.4 + 44000\lambda^2)u^2 + (34000\lambda^2)u + (-42.32 + 13600\lambda^2)$$

and

$$7) \qquad 0 = (-361.8 + 139500\lambda^2)v^4 + (45600\lambda^2)v^3 + (-127.8 + 36000\lambda^2)v^2 + (12000\lambda^2)v + (-20.7)$$

$$8) \qquad 0 = (-289.8)v^4 + (139500\lambda^2)v^3 + (-160.8 + 46500\lambda^2)v^2 + (36000\lambda^2)v + (-43.2 + 12000\lambda^2).$$

Note that (5)–(8) is very similar to the simultaneous eigenvalue problem that arises in the potential/antipotential or eigenvalue method in Section 1.

Now consider equations (5) and (6). For a fixed value of λ we have two polynomials in the single variable u. In general they will not have a root in common, but for certain λ's they will. These values are found by computing the resultant (an 8×8 determinant in this case) of the two polynomials (see Section 2b). This resultant (up to a constant factor) is:

$$0 = 4.13749 - 16328\lambda^2 + 2.66084 \times 10^7\lambda^4 - 2.22584 \times 10^{10}\lambda^6 + 9.42028 \times 10^{12}\lambda^8 - 1.5669 \times 10^{15}\lambda^{10}.$$

The only positive root is $\lambda = .0467053$.

Similarly taking the resultant of (7) and (8) we get another polynomial in λ which also has $\lambda = .0467053$ as a root! Taking this common value of λ, we find the common root of (5) and (6) to be

$$u = h_1/h_2 = 1.09535$$

and the common root of (7) and (8) to be

$$v = g_1/g_2 = 1.05002.$$

Normalizing h_2 to 1, we get

$$h_1 = 1.09535$$

$$h_2 = 1.00000$$

$$g_1 = 1.20256$$

$$g_2 = 1.14534.$$

Note that there is error in these computations and that the results are accurate only to five significant figures.

v. General Fire Allocation Methods.

The simple example presented in section 2.a.ii above can be easily generalized. For example, we could consider a more general Lanchester attrition model

$$\frac{db_i(t)}{dt} = -\sum_{k=1}^{n} p_{ik}c_{ik}r_k(t) - \sum_{\ell=1}^{n} p'_{i\ell}c'_{i\ell}b_i(t)r_\ell(t) \qquad i = 1,\ldots,m$$

$$\frac{dr_j(t)}{dt} = -\sum_{r=1}^{m} q_{jr}a_{jr}b_r(t) - \sum_{s=1}^{m} q'_{js}a'_{js}r_j(t)b_s(t) \qquad j = 1,\ldots,n$$

that includes both the square and linear laws (direct and indirect fire). The fire allocation factors p_{ik}, $p'_{i\ell}$, q_{jr}, and q'_{js} must satisfy

$$\sum_{j=1}^{n} q_{ji} + \sum_{j=1}^{n} q'_{ji} = 1 \quad \text{for} \quad i = 1,\ldots,m$$

and

$$\sum_{i=1}^{m} p_{ij} + \sum_{i=1}^{m} p'_{ij} = 1 \quad \text{for} \quad j = 1,\ldots,n.$$

These factors can be functions of numerous parameters including the kill rates a_{ji}, a_{ji}, c_{ij}, and c'_{ij}, the force levels $b_i(t_0)$, $r_j(t_0)$ at the current time $t = t_0$, and other constant factors (for example the analogues of the availability factors in ATCAL). Most importantly they should involve the unknown weapon values, h_i and g_j, of the blue weapons of type i and the red weapons of type j, respectively. These values h_i, g_j, once determined, should be interpreted as values at the current time $h_i(t_0)$, $g_j(t_0)$.

We are assuming here that each weapon type mounts one weapon. If this is not the case, then that particular type must appear more than once on the right hand side of our attrition equations. Moreover, if a particular weapon is not capable of both direct and indirect fire, it would not appear in both places, and the corresponding allocation factors would not appear. For example, the system below:

$$\begin{aligned}
\frac{db_1}{dt} &= -p_{11}a_{11}r_1 - p'_{11}a'_{11}b_1r_1 - p'_{12}a'_{12}b_1r_2 \\
\frac{db_2}{dt} &= -p_{21}a_{21}r_1 - p'_{21}a'_{21}b_2r_1 - p'_{22}a'_{22}b_2r_2 \\
\frac{dr_1}{dt} &= -q_{11}c_{11}b_1 - \tilde{q}_{11}\tilde{c}_{11}b_1 + q'_{12}c'_{12}r_1b_2 \\
\frac{dr_2}{dt} &= -q_{21}c_{21}b_1 - \tilde{q}_{21}\tilde{c}_{21}b_1 + q'_{22}c_{22}r_2b_2
\end{aligned}$$

represents a red type, r_1, that mounts a weapon capable of both direct and indirect fire, a red type, r_2, that mounts an indirect fire weapon, a blue type, b_1, that mounts two separate direct fire weapons, and a blue type, b_2, that is an indirect fire weapon.

As always, we assume that a weapon's value is proportional to the rate at which it destroys opposing value; so that

$$\begin{aligned}
\lambda h_1 &= \sum_{j=1}^{n} q_{j1}a_{j1}g_j + \sum_{j=1}^{n} q'_{j1}a'_{j1}g_jr_j(t_0) \\
&\vdots \\
\lambda h_m &= \sum_{j=1}^{n} q_{jm}a_{jm}g_j + \sum_{j=1}^{n} q'_{jm}a'_{jm}g_jr_j(t_0)
\end{aligned}$$

$$\lambda g_1 = \sum_{i=1}^{m} p_{i1} c_{i1} h_i + \sum_{i=1}^{m} p'_{i1} c'_{i1} h_i b_i(t_0)$$

$$\vdots$$

$$\lambda g_n = \sum_{i=1}^{m} p_{in} c_{in} h_i + \sum_{i=1}^{m} p'_{in} c'_{in} h_i b_i(t_0).$$

This is a system of $m+n$ equations in $m+n$ variables $h_1, \ldots, h_m$, $g_1, \ldots, g_n$, treating λ as a parameter in the coefficients. It is non-linear in general, because the expressions for q_{jr}, q'_{js}, p_{ik}, and $p'_{i\ell}$ will involve $h_1, \ldots, h_m$, $g_1, \ldots, g_n$. Moreover the equations should be homogeneous, since scaling all the values by the same amount should not cause a change in the allocation of fire, because the relative values will remain the same. To solve systems of this sort see 2b below. Solving the system in this case means finding those λ for which the system has a non-trivial solution. Again this is done with resultants.

Fire allocation methods are not limited to use with Lanchester attrition models. Any attrition model is a candidate for this method as long as we can link attrition rates, values, and allocation of fires as we have done above.

b. Resultants and Non-linear Equation Solving Techniques

This section is meant to serve as an introduction to resultants. Simple examples are presented which illustrate each technique. The reader should have little difficulty in applying the various methods to any particular problem.

Resultants are used to solve systems of non-linear polynomial equations, to determine whether or not solutions exist, or to reduce a given system to one with fewer variables and/or fewer equations.

The typical input will be a system of m equations in n-variables:

$$f_1(x_1, \ldots, x_n) = 0$$

$$\vdots$$

$$f_m(x_1, \ldots, x_n) = 0.$$

Each equation has an associated degree $d_i \geq 1$. Recall that $f_i(x_1, \ldots, x_n)$ has degree d_i if all monomials $x_1^{e_1} x_2^{e_2} \ldots x_n^{e_n}$ appearing in f_i have $\sum_{i=1}^{n} e_i \leq d_i$ and at least one monomial

has $\sum_{i=1}^{n} e_i = d_i$. Example: $f(x_1, x_2, x_3) = 3x_1^2 x_3 + 4x_1 x_2 - x_2 + 7x_3 - 1$ has degree $d = 3$. The integers $m, n, d_1, \ldots, d_m$ are important indicators of the specific resultant that will need to be employed.

We remark that the coefficients of the f_i may involve other unknown parameters. For example in our discussion of the fire allocation method, we constructed a simple example (2.a.ii) that led to a system of four homogeneous quadratic (degree 2) equations in four variables g_1, g_2, h_1, h_2 whose coefficients involved a parameter λ.

The typical output of a resultant depends on the number of equations versus the number of variables. It is important to note that when all the equations are homogeneous, we count the number of variables as being one less than the actual number. In this case we are interested in solutions other than the trivial solution (all zeros). There are two essentially different cases:

Case 1: $m > n$ (overdetermined).

This is the case where we have more equations than unknowns, and where we generally expect to have no solutions. The resultant will be a system of equations (one equation when $m = n + 1$) in the symbolic coefficients of the f_i, that has the following property: when we substitute the specific coefficients of the f_i, we will get zero in every equation in the resultant system if and only if the original overdetermined system has a solution.

Case 2: $m \leq n$ (exact and undetermined).

In this case the number of equations is less than or equal to the number of variables, and we expect to have solutions. In fact, if we allow complex solutions and solutions at infinity, we are guaranteed to have solutions.

Of course, only when $m = n$ do we expect a finite number s of solutions. Bezout's Theorem then provides a count of $s = d_1 d_2 \ldots d_m$ solutions (counting complex solutions, solutions at infinity, and counting with appropriate multiplicities). Unfortunately, the possibility also exists (even when $m = n$), that there will be an infinite number of solutions.

In general, for $m \leq n$, the resultant will be one equation in $n - m + 1$ of the variables. In effect, the resultant makes it possible for us to eliminate $m - 1$ of the variables. For example, if we choose to eliminate $x_{n-m+2}, \ldots, x_n$, then the resultant R will be a polynomial $R(x_1, \ldots, x_{n-m+1})$. If $(\alpha_1, \ldots, \alpha_{n-m+1})$ is a solution to $R = 0$, then there will exist values

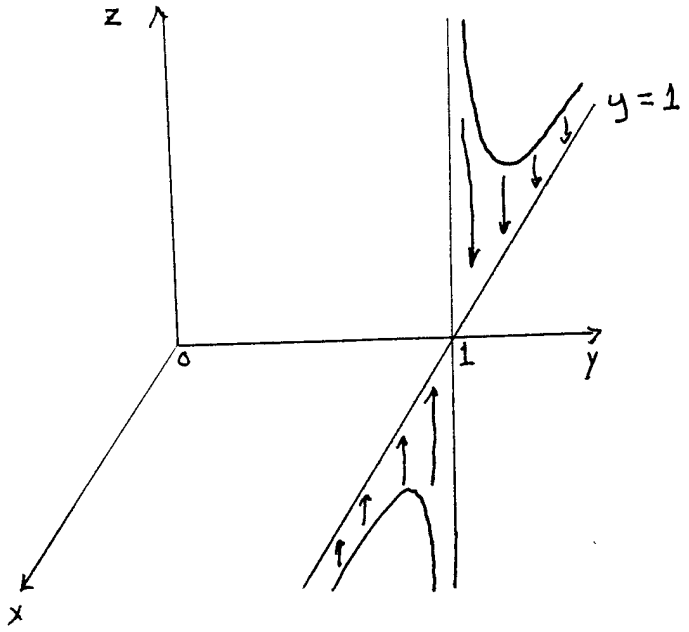
$\alpha_{n-m+2}, \ldots, \alpha_n$ such that $(\alpha_1, \ldots, \alpha_{n-m+1}, \alpha_{n-m+2}, \ldots, \alpha_n)$ is a solution to the original system. (One must be a little careful here. The system should be modified to make it homogeneous with respect to $x_{n-m+2}, \ldots, x_n$ by adding appropriate powers of a variable w. The values $\alpha_{n-m+2}, \ldots, \alpha_n$ should then be regarded as points $(\alpha_{n-m+2} : \ldots : \alpha_n : 1)$ in projective $m-1$ space $\mathbf{P}^{m-1}$. We must allow for the possibility that this point will be at infinity, where $w = 0$. In that case, a solution to $R = 0$ would not necessarily give rise to a solution of the original system.

Example: Consider the system of $m = 2$ equations in $n = 3$ variables: $4xyz - 1 = 0$ and $y + xz - 1 = 0$. The resultant eliminating z is $R(x, y) = x(4y^2 - 4y + 1)$. When $x = 0$ we will have $R = 0$, but clearly our system has no solution when $x = 0$. However, homogenizing with respect to z gives the system

$$4xyz - w = 0 \quad \text{and} \quad (y-1)w + xz = 0.$$

Now when we look at the condition $x = 0$ we find that $(z, w) = (1 : 0)$ is a solution. This is a point at infinity.

Notice that we also have solutions to $R = 0$ when $x \neq 0$ by taking $y = \frac{1}{2}$. This yields $z = \frac{1}{2x}$. Geometrically the solution set is a hyperbola in the plane $y = 1/2$ in space. The resultant "projects" that hyperbola to the line $y = 1/2$ in the xy-plane, except that $(x, y) = (0, 1/2)$ is not hit.



□

In this context (the underdetermined case) the resultant can be viewed as a projection of the nominally $n-m$ dimensional locus of solutions in $\mathbf{R}^n$ to an $n-m$ dimensional locus (hypersurface) in $\mathbf{R}^{n-m+1}$. Note that in our example $n=3$, $n=2$, and we are projecting the one-dimensional locus of solutions in $\mathbf{R}^3$ to a one-dimensional locus in $\mathbf{R}^2$ which is described by one equation $y-1/2=0$.

Our discussion in this section begins with the first major distinction in methods, namely the one based on the number of variables n. The case $n=1$ of a single variable is discussed first). We then move onto the multivariate case $n \geq 2$.

i. Resultants of Polynomials in One Variable

Given two positive integers $r, s \geq 1$ and two polynomials in one variable

$$f(x) = a_r x^r + \cdots + a_1 x + a_0 \qquad g(x) = b_s x^s + \cdots + b_1 x + b_0$$

of degree less than or equal to r and s respectively, we define their resultant $R_{r,s}(f,g)$ by

Sylvester's formula:

$$R_{r,s}(f,g) = \det \begin{pmatrix} a_0 & a_1 & & \dots & a_r & 0 & \dots & & 0 \\ 0 & a_0 & a_1 & \dots & \dots & a_r & 0 & \dots & 0 \\ \vdots & & & & & & & & \\ 0 & 0 & \dots & 0 & a_0 & a_1 & & \dots & a_r \\ b_0 & b_1 & \dots & & b_{s-1} & b_s & 0 & \dots & 0 \\ 0 & b_0 & b_1 & & \dots & & b_s & \dots & 0 \\ \vdots & & & & & & & & \\ 0 & 0 & \dots & & 0 & b_0 & b_1 & \dots & b_s \end{pmatrix}$$

which is the determinant of an $r+s$ by $r+s$ matrix with s rows involving the a's and r rows involving the b's. (Note that the a's and b's can be functions of other parameters, e.g. λ in our fire allocation examples; see page 27 for a case where $r=4$, $s=4$.)

For example the resultant of two quadratic equations is:

$$R_{2,2}(a_2x^2 + a_1x + a_0, b_2x^2 + b_1x + b_0) =$$

$$a_0^2b_2^2 + a_0a_2b_1^2 - a_0a_1b_1b_2 + a_1^2b_0b_2 - a_1a_2b_0b_1$$

$$+a_2^2b_0^2 - 2a_0a_2b_0b_2.$$

Note that each monomial in the resultant has total degree $r+s$ and that it is bihomogeneous of bidegree (s,r) in the a's and b's respectively.

Basic properties of the resultant $R_{r,s}(f,g)$.

1) Relationship to Common Roots

   $R_{r,s}(f,g) = a_r^s b_s^r \prod_{i,j}(x_i - y_j)$ where $x_1, \dots, x_r$ are the roots of f and $y_1, \dots, y_s$ are the roots of g. (Here we are assuming $a_r \neq 0$ and $b_s \neq 0$.) Thus $R_{r,s}(f,g)$ will be zero if and only if f and g have a root in common. When the a's and b's involve other parameters, $R_{r,s}(f,g)$ will be a function of those parameters. Setting $R_{r,s}(f,g) = 0$ will then tell us which values of the parameters cause f and g to have a root in common.

2) Irreducibility

   $R_{r,s}(f,g) \in \mathbf{Z}[a_0, \dots, a_r, b_0, \dots, b_s]$ is irreducible, i.e. the resultant is an irreducible polynomial with integer ($\mathbf{Z}$) coefficients in $(r+1)(s+1) = rs + s + r + 1$ variables. (Here we are treating the coefficients of f and g as symbolically, i.e. as independent variables.)

3) Symmetry

$R_{r,s}(f,g) = (-1)^{rs} R_{s,r}(g,f)$

4) Factorization

$R_{r_1+r_2,s}(f_1 f_2, g) = R_{r_1,s}(f_1, g) R_{r_2,s}(f_2, g)$.

ii. Resultant Methods for Systems of Polynomial Equations in Several Variables

The linear algebra techniques discussed in this section can be used to solve systems of polynomial equations in several variables. If there are only two equations, then the Sylvester technique (discussed above) can be employed, by treating all but one variable as part of the coefficients. However, when the number of equations exceeds two, the Sylvester approach can be misleading. For example, taking the equations two at a time using the Sylvester determinant can lead the user to the conclusion that there is a common solution, when in fact, there are no common solutions for the system of equations taken as a whole.

What it means to "solve" a given set of polynomial equations depends upon the number of variables and the number of equations. Assuming the equations are inhomogeneous, let "n" be the number of variables and "m" be the number of equations. The expected dimensionality of the set of solutions is $n - m$ when viewed over the complex numbers. For example, if there are three equations ($m = 3$) and five variables ($n = 5$), then the space of solutions is expected to have dimension $n - m = 5 - 3 = 2$. Geometrically, the solutions are surfaces. Sometimes, however, components of excess dimension occur in the space of solutions. These are geometric loci of higher dimension than the expected dimension. They occur because, in a very loose sense, the equations have certain dependencies.

Finally a word about homogeneous equations. Recall that a set of polynomial equations is considered homogeneous if in each equation, all the terms have the same degree. If this is not the case, even for only one of the equations, the set is regarded as inhomogeneous. For systems of homogeneous equations, the number n of variables should be taken as one less than the actual number of variables when computing expected dimensions.

The Macaulay Resultant and the U-Resultant

The Macaulay resultant is the ratio of two determinants formed from the coefficients of the given polynomials in a manner to be described later in this section. If the number

of equations exceeds the number of variables by one ($n - m = -1$), then the Macaulay resultant tests whether or not a common solution exists. (For systems of homogeneous equations where the number of equations equals the number of variables, the expected dimension is still -1, and the Macaulay resultant tests for a non-trivial common solution, i.e. a solution other than $(0, \ldots, 0)$.)

If there are as many inhomogeneous equations as unknowns ($n - m = 0$), then the equations can often be solved by adding the U-equation (explained later in this section) to the homogenized set and forming the Macaulay resultant. The Macaulay resultant is then called the U-resultant.

In some cases, however, there will be an excess component which masks some or all of the desired solutions. In this case Canny's Generalized Characteristic Polynomial (GCP) approach is useful (see [11]).

In order to illustrate the various methods, the following system of three polynomial equations will be used:

$$f_1 = y - 3x + 5 = 0$$

$$f_2 = x^2 + y^2 - 5 = 0$$

$$f_3 = y - x^3 + 3x^2 - 3x + 1 = 0.$$

Here we have three inhomogeneous equations in two variables ($n - m = 2 - 3 = -1$). The multiresultant techniques described below can be used to test for the existence of a solution.

Step 1: Homogenization

The equations must first be homogenized. This is done by adding a third variable, z. Specifically x is replaced by x/z and y is replaced by y/z, and the factors of z are cleared from the denominators. In the above example this leads to three equations

$$f_1 = y - 3x + 5z = 0$$

$$f_2 = x^2 + y^2 - 5z^2 = 0$$

$$f_3 = yz^2 - x^3 + 3x^2z - 3xz^2 + z^3 = 0.$$

This is the homogenized version of the original system.

Step 2: Degree Determination

Each of the multiresultants being considered involves the coefficients of various mononomials that appear in the equations of the system. The variables involved in the monomials are the set of variables that appear in the homogeneous form of the polynomial equations. For example, the homogeneous polynomial equations above have the variables x, y, and z. All the monomials in a given equation are constrained to have the same degree because we have homogenized. The "overall degree" of the system is determined from the degrees of the individual homogeneous equations by the following rule:

$$d = 1 + \sum_{i=1}^{m} (d_i - 1)$$

where

m = the number of equations

d_i = the degree of the "ith" equation.

For the homogeneous polynomials above (f_1, f_2, and f_3) the degrees are:

Equation	Degree
f_1	$d_1 = 1$
f_2	$d_2 = 2$
f_3	$d_3 = 3.$

Therefore,

$$d = 1 + (1 - 1) + (2 - 1) + (3 - 1) = 4.$$

(As another example, in our simple fire allocation model we had four homogeneous quadratic equations in the weapon values g_1, g_2, h_1, h_2 – this yields $d = 5$.)

Step 3: Matrix Size Determination

Each of the multiresultants to be discussed involves the ratio of two determinants. The numerator is the determinant of a matrix, the formation of which will be discussed in subsequent sections. The denominator determinant is formed from a submatrix of the numerator matrix.

The number of variables in the inhomogeneous equations is n. Since one additional variable has to be added to homogenize the equations, the number of variables in the

homogeneous equations is $n+1$. The size of the numerator matrix equals the number of monomials in the $n+1$ variables that have overall degree d (discussed in the previous section).

$$\text{Numerator Matrix Size} = \binom{n+d}{d}.$$

For the three polynomial equations (f_1, f_2, f_3) we have already calculated that $d = 4$. Since the original set of inhomogeneous variables consisted of x and y, n equals 2. Thus for our example,

$$\text{Numerator Matrix Size } = \binom{2+4}{4} = \binom{6}{4} = \frac{6!}{(2!)(4!)} = 15$$

i.e. it is a 15×15 matrix. (In the fire allocation case mentioned above the numerator size would be $\binom{8}{5} = 56$ which is rather large.)

Step 4: Determining "Big" vs. "Small" Exponents

A few of the 15 monomials involving the variables x, y, and z with an overall degree of 4, include:

(1) yz^3

(2) x^2y^2.

In the next step, we will discuss whether certain monomials are reduced. This will be determined by whether the exponents are "big" or "small". In this step we discuss how "bigness" is defined.

Each variable will be associated with a particular equation. For example the first variable, x, will be associated with the first equation, f_1. The second variable, y, will be associated with the second equation, f_2, etc. The degrees of the associated equations define "bigness" for the exponents of that variable. Specifically, since d_1 (the degree of f_1) is 1, if the exponent of x is greater than or equal to 1, it is considered big. Since $d_2 = 2$, whenever the exponent of y is greater than or equal to 2, it is considered big. The degree of f_3 is 3, therefore, whenever the exponent of z is greater than or equal to 3, it is considered big.

For example, consider the monomial numbered (1), namely, yz^3. The exponent of y is 1. This is *less than* d_2, and is considered *small*. The exponent of z is 3. This is *equal to*

d_3, and is therefore *big*. On the other hand, if we consider the monomial numbered (2), namely, x^2y^2. The exponent of x is 2. This is *greater than* d_1, and is *big*. The exponent of y is 2. This is *equal to* d_2, and is *big*.

Step 5: Determining the Reduced Monomials

If for a particular monomial of degree d the exponent of *only one* variable is big, the monomial is said to be reduced. In the previous step only monomial (1) is reduced. For monomial (1) only the exponent of z is big; whereas for (2), both the exponent of x and the exponent of y are big. Thus monomial (2) is not reduced.

The Macaulay Resultant

The Macaulay Resultant is the ratio of two determinants. The numerator is the determinant of a matrix which we will call the A matrix. The denominator is the determinant of a matrix which we will call the M matrix

$$R = \frac{\det |A|}{\det |M|}.$$

Step 6: Creating the A Matrix:

We have discussed above how the size of the A matrix is determined. In this step we will show how the matrix elements are obtained.

Each row and column of the matrix should be thought of as being labeled by one of the monomials of degree d. Recall that for f_1, f_2, and f_3 there were 15 possible monomials of degree 4 in x, y, z, and therefore the A matrix would be 15×15.

There are three rules for determining the elements of the A matrix. After presenting the rules, the example involving f_1, f_2, and f_3, will be used to illustrate the process. The reader may find it helpful to read the example simultaneously with the rules.

Rules for the elements of each column:

(1) Search the monomial labeling that column from left to right for the *first* variable with a big exponent. Such a variable must exist. Call it the marker variable.

(2) Form a new polynomial from the polynomial associated with this marker variable by multiplying the associated polynomial by the monomial and dividing by the marker variable raised to the degree of the associated polynomial.

(3) The coefficients of the new polynomial are the elements of the columns. Each coefficients goes in the row labeled by the monomial it multiplies. All the other rows get zeroes.

Recall that for our example system of equations f_1, f_2, f_3 there are 15 monomials of degree 4 that can be formed from x, y, and z. Two of these were considered above, namely yz^3 and x^2y^2.

(a) For the column labeled by yz^3:

(1) The first variable with a big exponent is z, so z is the marker variable.

(2) The polynomial associated with z is f_3. Multiply f_3 by the monomial yz^3, and divide this product by z^3.

$$\frac{f_3(yz^3)}{z^3} = \frac{(yz^2 - x^3 + 3x^2z - 3xz^2 + z^3)(yz^3)}{z^3} = y^2z^2 - x^3y + 3x^2yz - 3xyz^2 + yz^3.$$

(3) The coefficient of y^2z^2 is $+1$. Therefore the element of the row labeled y^2z^2 is $+1$. The coefficient of x^3y is -1. Therefore the element of the row labeled x^3y is -1. The coefficient of x^2yz is $+3$. Therefore the element of the row labeled x^2yz is $+3$. The coefficient of xyz^2 is -3. Therefore the element of the row labeled xyz^2 is -3. The coefficient of yz^3 is $+1$. Therefore the element of the row labeled yz^3 is $+1$. All other entries in the column are zero.

(b) For the column labeled by x^2y^2:

(1) The first variable with a big exponent is x, so x is the marker variable.

(2) The polynomial associated with x is f_1. Multiply f_1 by the monomial x^2y^2, and divide this product by x.

$$\frac{f_1(x^2y^2)}{x} = \frac{(y - 3x + 5z)(x^2y^2)}{x} = xy^3 - 3x^2y^2 + 5xy^2z.$$

(3) The coefficient of xy^3 is $+1$. Therefore the element of the row labeled xy^3 is $+1$. The coefficient of x^2y^2 is -3. Therefore the element of the row labeled x^2y^2 is -3. The coefficient of xy^2z is $+5$. Therefore the element of the row labeled xy^2z is $+5$.

When all the columns are determined, the A matrix in our example takes the form:

A MATRIX

			x^4	$x^3 y$	$x^3 z$	$x^2 y^2$	$x^2 yz$	$x^2 z^2$	xy^3	$xy^2 z$	xyz^2	xz^3	y^4	$y^3 z$	$y^2 z^2$	yz^3	z^4
x^4			−3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
x^3	y		1	−3	0	0	0	0	0	0	0	0	0	0	0	−1	0
x^3		z	5	0	−3	0	0	0	0	0	0	0	0	0	0	0	−1
x^2	y^2		0	1	0	−3	0	0	0	0	0	0	0	1	0	0	0
x^2	y	z	0	5	1	0	−3	0	0	0	0	0	0	1	0	3	0
x^2		z^2	0	0	5	0	0	−3	0	0	0	0	0	0	1	0	3
x	y^3		0	0	0	1	0	0	−3	0	0	0	0	0	0	0	0
x	y^2	z	0	0	0	5	1	0	0	−3	0	0	0	0	0	0	0
x	y	z^2	0	0	0	0	5	1	0	0	−3	0	0	0	0	−3	0
x		z^3	0	0	0	0	0	5	0	0	0	−3	0	0	0	0	−3
	y^4		0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
	y^3	z	0	0	0	0	0	0	5	1	0	0	0	1	0	0	0
	y^2	z^2	0	0	0	0	0	0	0	5	1	0	−5	0	1	1	0
	y	z^3	0	0	0	0	0	0	0	0	5	1	0	−5	0	1	1
		z^4	0	0	0	0	0	0	0	0	0	5	0	0	−5	0	1

The determinant of the above A matrix is zero. If the M matrix is nonzero, this would imply that the system has a solution.

Step 7: Creating the M Matrix:

The denominator of the Macaulay Resultant is the determinant of the M matrix. The M matrix is a submatrix of the A matrix. It consists of the elements which have row and column monomial labels which are *not reduced*. Recall that a monomial is not reduced if it has more than one variable with a big exponent.

The size of the M matrix equals the size of the A matrix minus D, where

$$D = \sum_{i=1}^{m} \prod_{i \neq j} d_j.$$

In our example,

$$D = d_2 d_3 + d_1 d_3 + d_1 d_2 = (2)(3) + (1)(3) + (1)(2) = 11.$$

So that the size of the M matrix is $15 - 11 = 4$. The actual M matrix for f_1, f_2 and f_3 is:

M MATRIX

	x^2y^2	xy^3	xy^2z	xz^3
x^2y^2	-3	0	0	0
xy^3	1	-3	0	0
xy^2z	5	0	-3	0
xz^3	0	0	0	-3

The determinant of this M matrix yields a value of 81. Since the A matrix was zero, the Macaulay Resultant is zero which implies that there is a solution to our system. A quick plot of the three polynomials (f_1, f_2, and f_3) confirms that there is a common point at $x = 2$ and $y = 1$.

Sometimes both the A matrix and the M matrix have zero determinant. This indeterminacy can often be circumvented if the polynomials are first written with symbolic coefficients. The determinants of the A and M matrices are then computed, and polynomial division is performed. At that point the symbolic coefficients are replaced by their numerical values to check if the resultant is zero. Since one does not know ahead of time whether or not this "division by zero" condition will arise, the symbolic coefficient approach is the best strategy. It is also often sufficient to treat just a subset of the coefficients symbolically – sometimes as few as a single symbolic coefficient will remove the indeterminacy.

The U-Resultant

For problems with as many inhomogeneous equations as variables, the U resultant can often be used to solve for the point solutions. The three polynomial equations f_1, f_2, f_3 do not satisfy these conditions since there are three equations in two inhomogeneous variables, x and y. However, if we take just the first two equations, namely f_1 and f_2, we would have a system with as many equations as variables.

The given equations must first be homogenized. This adds one additional variable. We then add one additional equation to the system. This equation is called the U-equation. If x and y are the given variables and z is the homogenizing variable, then the U equation

takes the form:

$$u_1 x + u_2 y + u_3 z = 0.$$

The Macaulay Resultant, R, is then computed for these $m+1$ equations, treating the u_i as symbolic coefficients. The result is called the U-resultant. Notice that R will be a polynomial in the u_i's and the coefficients of the original equations.

After R is determined, it is then factored into linear factors. For each linear factor there is a point solution of the original system of equations. The coordinates of each solution are determined as ratios of the coefficients of the u's. The denominator is always the coefficient of the u_i associated with the homogenizing variable. In our example, this is the coefficient of u_3. Thus

$$x = \frac{\text{coeff. of } u_1}{\text{coeff. of } u_3} \quad y = \frac{\text{coeff. of} u_2}{\text{coeff. of } u_3}.$$

For example, if a linear factor turned out to be

$$u_1 - u_2 - u_3,$$

then the coordinates of the associated solution would be

$$x = \frac{+1}{-1} = -1 \qquad y = \frac{-1}{-1} = +1.$$

As mentioned above, we can use the U-resultant to solve f_1 and f_2 for x and y ($n-m = 0$). In this example, we will also demonstrate the symbolic approach alluded to in the previous section. Recall that the homogenized form of f_1 and f_2 is:

$$f_1 = y - 3x + 5z = 0$$

$$f_2 = x^2 + y^2 - 5z^2 = 0.$$

Rewriting these two equations with symbolic coefficients and including the U-equation yields:

$$f_1 = a_1 x + b_1 y + c_1 z = 0$$

$$f_2 = a_2 x^2 + b_2 y^2 + c_2 z^2 = 0$$

$$U = u_1 x + u_2 y + u_3 z = 0$$

where, $a_1 = -3; b_1 = 1; c_1 = 5; a_2 = 1; b_2 = 1; c_2 = -5$.

The U-resultant is calculated in the same way as the Macaulay resultant, i.e. with the A matrix and the M matrix, except now we are using symbolic coefficients.

A MATRIX

	x^2	xy	xz	y^2	yz	z^2
x^2	a_1	0	0	a_2	0	0
xy	b_1	a_1	0	0	u_1	0
xz	c_1	0	a_1	0	0	u_1
y^2	0	b_1	0	b_2	u_2	0
yz	0	c_1	b_1	0	u_3	u_2
z^2	0	0	c_1	c_2	0	u_3

The corresponding M matrix is a single element, namely a_1.

The determinant of M is divided into the determinant of A to obtain the U-resultant. Finally, the symbolic coefficients are replaced by their numeric equivalents. (This could have been done from the outset, unless a_1 had been zero.) The result is

$$10(u_1 - 2u_2 + u_3)(2u_1 + u_2 + u_3).$$

This yields two solutions:

Solution #1:

$$x = \frac{\text{coeff. of } u_1}{\text{coeff. of } u_3} = \frac{+1}{-1} = +1 \quad \text{and} \quad y = \frac{\text{coeff. of } u_2}{\text{coeff. of } u_3} = \frac{-2}{+1} = -2.$$

Solution #2:

$$x = \frac{\text{coeff. of } u_1}{\text{coeff. of } u_3} = \frac{+2}{+1} = +2 \quad \text{and} \quad y = \frac{\text{coeff. of } u_2}{\text{coeff. of } u_3} = \frac{+1}{+1} = +1.$$

We remark that the U-resultant will be identically zero and give no information if the set of common solutions contains a component of excess dimension one or more. Moreover, this component may be at infinity where the homogenizing variable is zero.

The GCP Approach

The Generalized Characteristic Polynomial (GCP) approach avoids the problem of components of excess dimension in the space of solutions. It can be used together with the U-resultant which was discussed above. If the U-resultant leads to an indeterminant

(0/0) form even when symbolic coefficients are used, an "excess" solution exists. The GCP takes the form

$$R = \frac{\det|A - sI|}{\det|M - sI|} \text{ evaluated at } s = 0 \text{ (after division)}$$

where A and M are the matrices defined above, s is a perturbation parameter, and I is the identity matrix.

One way to carry out the above operation is the following:

(1) Set up the A matrix (as described above). Subtract an s along the diagonal. Evaluate the determinant. Retain the coefficient of the lowest surviving power of s.

(2) Repeat (1) for the M matrix.

(3) Divide the result of (1) by the result of (2).

All of these multiresultant techniques have one thing in common. They require that there be one more equation than variable, $n - m = -1$. If there are as many equations as variables $n - m = 0$, then the U equation is added and the effective situation is again $n - m = -1$. If there are more variables than equations ($n - m$ is a positive integer), then enough of these variables must be regarded as parameters in the coefficients, so that effectively $n - m = -1$. Geometrically this amounts to projecting the locus of solutions to a hypersurface in a lower dimensional Euclidean space. Finally, if the number of equations exceeds the number of variables by more than one ($n - m \leq -2$), then some technique other than the above Macaulay resultant (e.g. a systems of resultants) must be employed to determine if a solution exists.

c. Similarities with and Differences from other Methods.

The fire allocation method is by far the most general method we discuss in this paper. It couples fire allocation, attrition or kill rates, and weapon values in a natural way, and leads to a system of non-linear equations in the weapon values and the proportionality constant λ which has units of $(\text{time})^{-1}$.

This system can in turn be written as two systems of non-linear equations – one for the blue values and one for the red values – linked by the common "eigenvalue" λ^2. The

similarity of this result with the simultaneous eigenvalue problem arising in the eigenvalue or potential/antipotential method is quite striking. Unfortunately, we have not yet been able to prove an existence or uniqueness result for these non-linear systems. In the potential/antipotential case, the existence and uniqueness is assured under some mild hypotheses. We have however done some numerical experiments that have led us to conjecture that these non-linear systems do indeed have solutions and that they are unique.

The major drawback with the fire allocation models discussed above is that solving coupled system of non-linear equations is not an easy task. The resultant methods we suggest grow exponentially in size with the number of weapon types. Dealing with 50 plus types per side would require very large computations. Perhaps some blend of symbolic and numeric techniques could be employed to make such a computation. Canny [11] has successfully explored such an approach in computing very large resultants.

Another drawback is that the pure kill rates (the a_{ji} and c_{ij} that appear in our example) need to be known in advance. The best approach here is to make use of the results of a high resolution low level simulation (e.g. COSAGE) where fire allocation can be more or less determined, so that along with the killer-victim scoreboards, the pure attrition coefficients can be estimated.

Finally, on the positive side, we believe that the iterative scheme we have outlined for coupling attrition, fire allocation, and weapon values is potentially quite useful. It certainly warrants further investigation.

3. Weapon Importances in ATCAL

a. The Methods

ATCAL is a calibrated attrition model used to determine attrition in a theater level simulation called CEM at the U.S. Army Concepts Analysis Agency. Certain parameters are determined (calibrated) using the results of a high resolution, low level (division level for example) simulation called COSAGE. Multiple COSAGE runs are done with a mix of all the weapons systems available in order to capture all possible interactions.

ATCAL weapons importances are calculated from entries in the killer-victim score-board. Let

$$a_{ij} = \frac{(\Delta M_j)_i}{M_j \Delta T} \quad \text{and} \quad b_{ji} = \frac{(\Delta N_i)_j}{N_i \Delta T}$$

for $1 \leq i \leq n$ and $1 \leq j \leq m$. Here $(\Delta M_j)_i$ is the number of type j killed by opposing type i during the engagement, ΔT is the length of the engagement, and M_j is the initial number of type j present at the beginning of the engagement. Thus a_{ij} is the fraction of type j killed by opposing type i per unit time. Likewise b_{ji} is the fraction of type i killed by opposing type j per unit time. Note that $\sum_{k=1}^{n} a_{kj}$ equals the fraction of j killed per unit time and $\sum_{s=1}^{m} b_{si}$ equals the fraction of i killed per unit time.

The weapon importances X_i, $i = 1, \ldots, n$ and Y_j, $j = 1, \ldots, m$ are defined by:

$$X_i^3 = \sum_{j=1}^{m} \frac{(a_{ij})^3}{\sum_{k=1}^{n} a_{kj}} Y_j \quad \text{for} \quad i = 1, \ldots, n$$

$$Y_j^3 = \sum_{i=1}^{n} \frac{(b_{ji})^3}{\sum_{s=1}^{m} b_{si}} X_i \quad \text{for} \quad j = 1, \ldots, m$$

$$X_i \geq 0$$

$$Y_j \geq 0.$$

Notice that $X_i = 0$ for all i and $Y_j = 0$ for all j is a solution to the above system of equations. Clearly we want a non-trivial solution where not all these values are zero. Note also the X_i is the total importance of all weapons of that type as is Y_j.

In this system of non-linear equations the quantity

$$\frac{b_{ji}}{\sum_{s=1}^{m} b_{si}} = \frac{(\Delta N_i)_j}{\Delta N_i}$$

is the fraction of all kills of i that were caused by j. Likewise

$$\frac{a_{ij}}{\sum_{k=1}^{n} a_{kj}} = \frac{(\Delta M_j)_i}{\Delta M_j}$$

is the fraction of all kills of j that were due to i.

We will discuss the motivations advanced for these equations in section 3c. For now we content ourselves with showing one important property exhibited by this formulation of weapon importances, namely the so-called divisibility property. It says that if we divide a type into two types the total importance assigned to them does not change.

To prove this, suppose we divide N_i for type $i = 1$ into $N_1' + N_1'' = N_1$ and apportion losses and kills accordingly:

$$\frac{N_1'}{N_1} = r \quad \frac{N_1''}{N_1} = 1 - r$$

$$a_{1j}' = \frac{r(\Delta M_j)_1}{M_j \Delta T} = r a_{1j}$$

$$a_{1j}'' = (1-r) a_{1j}'$$

$$b_{j1}' = \frac{r(\Delta N_1)_j}{(r N_1)\Delta T} = b_{j1}$$

$$b_{j1}'' = b_{j1}.$$

Our new set of importance equations are

$$\tilde{X}_1'^3 = \sum_{j=1}^{m} \frac{(a_{1j}')^3}{\sum_{k=1}^{n} a_{kj}} \tilde{Y}_j = \sum_{j=1}^{m} \frac{(a_{1j})^3}{\sum_{k=1}^{n} a_{kj}} r^3 \tilde{Y}_j$$

$$\tilde{X}_1''^3 = \sum_{j=1}^{m} \frac{(a_{1j}'')^3}{\sum_{k=1}^{n} a_{kj}} \tilde{Y}_j = \sum_{j=1}^{m} \frac{(a_{1j})^3}{\sum_{k=1}^{n} a_{kj}} (1-r)^3 \tilde{Y}_j$$

$$\tilde{X}_i^3 = \sum_{j=1}^{m} \frac{(a_{ij})^3}{\sum_{k=1}^{n} a_{kj}} \tilde{Y}_j \qquad i = 2, \ldots, n.$$

(Note: $a'_{1j} + a''_{1j} + a_{2j} + \cdots + a_{nj} = a_{1j} + \cdots + a_{nj}$.)

$$\widetilde{Y}_j^3 = \frac{(b'_{j1})^3}{\sum_{j=1}^{m} b'_{s1}} \widetilde{X}'_1 + \frac{(b''_{1j})^3}{\sum_{s=1}^{m} b''_{s1}} \widetilde{X}''_1 + \sum_{i=2}^{n} \frac{(b_{ji})^3}{\sum_{s=1}^{m} b_{si}} \widetilde{X}_i \quad j = 1, \ldots, m.$$

(Note: $\frac{(b'_{j1})^3}{\sum_{s=1}^{m} b'_{s1}} = \frac{(b''_{j1})^3}{\sum_{s=1}^{m} b''_{s1}} = \frac{(b_{j1})^3}{\sum_{s=1}^{m} b_{s1}}$.)

Now if $(X_1, \ldots, X_n, Y_1, \ldots, Y_m)$ solves the original system then

$$(\widetilde{X}'_1, \widetilde{X}''_1, \widetilde{X}_2, \ldots, \widetilde{X}_n, \widetilde{Y}_1, \ldots, \widetilde{Y}_m) = (rX_1, (1-r)X_1, X_2, \ldots, X_n, Y_1, \ldots, Y_m)$$

solves the new system. Conversely, if $(\widetilde{X}'_1, \widetilde{X}''_1, \widetilde{X}_2, \ldots, \widetilde{Y}_m)$ satisfies the new system then $\frac{\widetilde{X}'_1}{r} = \frac{\widetilde{X}''_1}{1-r}$ and

$$(X_1, X_2, \ldots, Y_m) = \left(\frac{\widetilde{X}'_1}{r}, \widetilde{X}_2, \ldots, \widetilde{Y}_m \right)$$

satisfies the original equations. In particular

$$X_1 = \frac{\widetilde{X}'_1}{r} = \frac{\widetilde{X}''_1}{1-r}$$

so that

$$\widetilde{X}'_1 + \widetilde{X}''_1 = rX_1 + (1-r)X_1 = X_1.$$

This means that the total value of the prime and double prime weapon types equals the original total value X_1 for type $i = 1$. This proves the desired divisibility property.

Notice that there is nothing sacred in this proof about using cubes. Any power α works as long as the exponents match

$$X_i^\alpha = \sum_{j=1}^{m} \frac{(a_{ij})^\alpha}{\sum_{k=1}^{n} a_{kj}} Y_j \quad \text{for} \quad i = 1, \ldots, n$$

$$Y_j^\alpha = \sum_{i=1}^{n} \frac{(b_{ji})^\alpha}{\sum_{s=1}^{m} b_{si}} X_i \quad \text{for} \quad j = 1, \ldots, m.$$

This makes the choice of cubes seem somewhat arbitrary. We remark that the existence and uniqueness proof below for $\alpha = 3$ will also work for arbitrary α.

b. Existence and Uniqueness of Solutions to the ATCAL Weapon Importance Equations.

ATCAL has been criticized on a number of counts (see Anderson [56]), including the lack of any proof that its weapon importance equations always have a solution and that such a solution is unique. In this section we provide that proof. While this is largely a mathematical exercise, the proof yields two important insights. The first is an efficient algorithm, based on an iterative fixed point method, for computing the importances, and the second is the observation that there is a striking similarity between this proof and the fixed point proof of the Perron-Frobenius Theorem (see §1c) which is the key result in establishing existence and uniqueness for the so-called eigenvalue or potential/antipotential method of computing weapon scores. This observation leads to speculation that the eigenvalue scores and the ATCAL importances will suffer from similar defects.

We must establish our existence and uniqueness result for a system of polynomial equations of the form:

$$X_i^3 = \sum_{j=1}^{n} b_{ij} X_j \qquad i = 1, \ldots, n \tag{1}$$

where $n \geq 2$ and $b_{ij} \geq 0$ for every $1 \leq i, j \leq n$. We seek a solution $\vec{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathbf{R}^n_{\geq 0} - \{(0, \ldots, 0)^T\}$, that is, a solution with all $x_i \geq 0$ but not all $x_i = 0$. Let

$$\theta\colon \mathbf{R}^n_{\geq 0} - \{(0, \ldots, 0)^T\} \longrightarrow \mathbf{R}^n_{\geq 0} - \{(0, \ldots, 0)^T\}$$

be the mapping which sends $\vec{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$ to $\theta(\vec{x}) = \begin{pmatrix} x_1^{1/3} \\ \vdots \\ x_n^{1/3} \end{pmatrix}$.

In order to show uniqueness, we must make one assumption. Namely, we assume that the null space of the matrix $B = (b_{ij})$ intersects the positive orthant $\mathbf{R}^n_{\geq 0}$ only in the zero vector $(0, \ldots, 0)^T$. This means that $B\vec{x} \in \mathbf{R}^n_{\geq 0} - \{(0, \ldots, 0)^T\}$ for every $\vec{x} \in \mathbf{R}^n_{\geq 0} - \{(0, \ldots, 0)^T\}$ and is a very mild assumption given that all the entries of B are non-negative.

Now consider the standard $(n-1)$-simplex $\Delta^{n-1} = \{(x_1, \ldots, x_n)^T \in \mathbf{R}^n_{\geq 0}$ such that $\sum_{i=1}^{n} x_i = 1$, and $x_i \geq 0$ all $i = 1, \ldots, n\}$ in $\mathbf{R}^n_{\geq 0}$ and the map

$$\Delta^{n-1} \longrightarrow \Delta^{n-1}$$

$$\vec{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \longmapsto \frac{\theta(B\vec{x})}{\sigma(\theta(B\vec{x}))}$$

where σ: $\mathbf{R}^n \to \mathbf{R}$ sends $(x_1, \ldots, x_n)^T$ to $\sum_{i=1}^{n} x_i$. By the Brouwer fixed point theorem, there is a vector

$$\vec{w}_0 = \begin{pmatrix} w_1 \\ \vdots \\ w_n \end{pmatrix} \in \Delta^{n-1}$$

with

$$\vec{w}_0 = \frac{\theta(B\vec{w}_0)}{\sigma(\theta(B\vec{w}_0))}.$$

Setting $\chi = \sigma(\theta(B\vec{w}_0))$, we have

$$\chi\vec{w}_0 = \theta(B\vec{w}_0)$$

for a value $\chi > 0$. Equivalently,

$$\chi^3 \begin{pmatrix} w_1^3 \\ \vdots \\ w_n^3 \end{pmatrix} = B \begin{pmatrix} w_1 \\ \vdots \\ w_n \end{pmatrix}.$$

This is almost, but not quite, a solution to (1). If we scale $\vec{w}_0$ by λ, we observe that

$$\chi^3 \begin{pmatrix} (\lambda w_1)^3 \\ \vdots \\ (\lambda w_n)^3 \end{pmatrix} = \chi^3\lambda^3 \begin{pmatrix} w_1^3 \\ \vdots \\ w_n^3 \end{pmatrix} = \lambda^3 B \begin{pmatrix} w_1 \\ \vdots \\ w_n \end{pmatrix}$$

$$= \lambda^2 B \begin{pmatrix} \lambda w_1 \\ \vdots \\ \lambda w_n \end{pmatrix}.$$

If we now pick $\lambda = \sqrt{\chi^3}$, then

$$\vec{v}_0 = \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} = \begin{pmatrix} \lambda w_1 \\ \vdots \\ \lambda w_n \end{pmatrix} = \lambda\vec{w}_0$$

solves (1), i.e.

$$\begin{pmatrix} v_1^3 \\ \vdots \\ v_n^3 \end{pmatrix} = B \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}.$$

This establishes the existence of a non-negative, not identically zero solution to (1).

We now turn to uniqueness. By the argument above we know that

$$X_i^3 = \sum_{j=1}^{n} b_{ij} X_j \quad b_{ij} \geq 0 \quad i = 1, \ldots, n$$

has at least one solution (under a mild hypothesis on $B = (b_{ij})$), call it

$$\vec{v}_0 = \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}.$$

We will assume all $v_i > 0$. (For example one can throw out all weapons with zero value and the corresponding equations.)

Now set $Y_i = \frac{X_i}{v_i}$ to get a new system of equations

$$Y_i^3 = \sum_{j=1}^{n} c_{ij} Y_j \tag{2}$$

where $c_{ij} = \frac{b_{ij} v_j}{v_i^3} \geq 0$ and $\sum_{j=1}^{n} c_{ij} = 1$ for all $i = 1, \ldots, n$. This new system has $\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$ as a solution.

Suppose there is another solution to (2)

$$\vec{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \qquad y_i > 0$$

with some $y_i > 1$. Let i_0 be the index for which y_{i_0} is a maximum. We have

$$\begin{aligned} y_{i_0}^3 - y_{i_0} &= \sum_{j=1}^{n} c_{i_0 j} y_j - \left(\sum_{j=1}^{n} c_{i_0 j} \right) y_{i_0} \\ &= \sum_{j=1}^{n} c_{i_0 j} (y_j - y_{i_0}), \end{aligned}$$

but $y_{i_0}^3 - y_{i_0} > 0$ while $c_{i_0 j}(y_j - y_{i_0}) \leq 0$. This yields a contradiction. Thus if

$$\vec{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

is another solution, then all y_i are less than or equal to one and at least one y_i is less than one. (We continue to assume all $y_i > 0$.)

Let i_0 be the index for which y_{i_0} is a minimum. Then

$$y_{i_0} - y_{i_0}^3 = \sum_{j=1}^{n} c_{i_0 j}(y_{i_0} - y_j)$$

but $y_{i_0} - y_{i_0}^3 > 0$ while $\sum_{j=1}^{n} c_{i_0 j}(y_{i_0} - y_j) < 0$. □

We caution that this "proof' of uniqueness is, strictly speaking, incomplete. What we have shown is that we can't have two different solutions with all weapon importances being positive. But we can't rule out multiple solutions when some y_i are zero.

This problem with uniqueness breaking down is virtually identical to what happens in the eigenvalue method when multiple solutions occur. Moreover, our existence proof is virtually identical to the existence proof for the eigenvalue method given in Section 1. One is tempted to conclude that both the eigenvalue method and the ATCAL method will exhibit, at least qualitatively, similar behaviour. In particular both methods will suffer from similar defects in degenerate cases, and any criticism of the eigenvalue method is likely to apply equally to the method in ATCAL.

Finally the existence proof above provides a very stable iterative algorithm for solving the importance equations. Specifically, compute the $(i+1)st$ iterate by

$$\vec{x}_{i+1} = \frac{\theta(B\vec{x}_i)}{\sigma(\theta(B\vec{x}_i))}$$

starting with $\vec{x}_0 = (\frac{1}{n}, \ldots, \frac{1}{n})$ say. Once the fixed point is found, we scale as indicated in the proof to get a solution.

c. Problems and Similarity with the Eigenvalue Method.

The ATCAL weapon importance equations have a somewhat ad hoc feel to them. We were not able to find any derivation of them from first principles, as we can for the other methods. Moreover, as indicated in 3a, the cubic exponent is not the only exponent that will produce a satisfactory notion of value.

On the positive side, the importance equations reduce to the eigenvalue method when there is only one weapon per side:

$$X_1^3 = a_{12}^2 X_2$$

$$X_2^3 = a_{21}^2 X_1$$

where $a_{12} = \frac{(\Delta N_2)_1}{N_2 \Delta T} = \frac{\Delta N_2}{N_2 \Delta T}$ and $a_{21} = \frac{(\Delta N_1)_2}{N_1 \Delta T} = \frac{\Delta N_1}{N_1 \Delta T}$ are the fractional amounts killed per unit time. The solution to these equations is

$$X_1 = a_{12}^{3/4} a_{21}^{1/4}$$

$$X_2 = a_{12}^{1/4} a_{21}^{3/4}.$$

Thus the value of a single unit of type 1 is $V_1 = X_1/N_1$ and of type 2 is $V_2 = X_2/N_2$. Normalizing V_1 to 1 gives

$$\frac{V_2}{V_1} = \frac{X_2}{N_2} \cdot \frac{N_1}{X_1} = \frac{N_1}{N_2} \sqrt{\frac{a_{21}}{a_{12}}}.$$

Conversely the eigenvalue method for two weapons, one Red and one Blue, gives

$$CB \times VB = kVR \qquad k = \frac{(\Delta R)_B}{B \cdot \Delta T} = \frac{\Delta R}{B \Delta T}$$

$$CR \times VR = \ell VR \qquad \ell = \frac{(\Delta B)_R}{R \cdot \Delta T} = \frac{\Delta B}{R \Delta T}$$

where $CB = CR = \sqrt{\lambda}$ and where we normalize VB to 1. The simultaneous eigenvalue equations are

$$\lambda VB = k\ell\, VB$$

$$\lambda VR = k\ell\, VR$$

and we see that $\lambda = k\ell = a_{RB} a_{BR}$ where

$$a_{RB} = \frac{\Delta B}{B \Delta T} \qquad a_{BR} = \frac{\Delta R}{R \Delta T}.$$

We thus also get

$$VR \frac{\sqrt{\lambda}}{k} = \sqrt{\frac{\ell}{k}} = \frac{B}{R} \sqrt{\frac{a_{RB}}{a_{BR}}}$$

which is the same as ATCAL gave above (with indices 1 and 2 instead of B and R).

One odd feature of the ATCAL importances is that they depend on the units of time used. The unit of importance is $(\text{time})^{-1}$ which is an intensity (by analogy with Lanchester and/or the eigenvalue method). However, for relative importances this effect cancels out. Moreover there is a certain logical inconsistency, in that the ATCAL attrition model assumes exponential averaging, i.e. that attrition follows an exponentially decaying curve over time, while the importance equation are based on constant fractional loss per unit time. Why not incorporate the exponential decay into the model? In fact, one could compute fractional losses as a function of time using COSAGE data. This would give time dependent importances which could be integrated over the duration of the engagement to give dimensionless numbers for weapon values.

One possible derivation of the ATCAL importance equations is given below, but it is only heuristic, and is based on very poor assumptions about times to completion of various interactions. Recall that in the eigenvalue method:

$$CB \times VB_i = \sum_{j=1}^{n} K_{ij} VR_j$$

$$CR \times VR_j = \sum_{i=1}^{m} L_{ji} VB_i$$

where $CB = CR = \sqrt{\lambda} = I$ (intensity). In other words, a weapon's value is proportional to the rate at which it destroys opposing value (at a given instant in time). This leads to

$$CB \times VB_i \times B_i = \sum_{j=1}^{n} \frac{K_{ij} B_i \Delta T}{R_j \Delta T} \times VR_j \times R_j$$

or

$$CB \times TV_{B_i} = \sum_{j=1}^{n} \frac{(\Delta R_j)_i}{R_j \Delta T} \times TV_{R_j} \qquad \text{etc.}$$

where TV_{B_i} is the total value of Blue weapon's of type i, etc. This yields:

$$TV_{B_i} = \sum_{j=1}^{n} a_{ij} TV_{R_j} \left(\frac{1}{CB}\right)$$

and

$$TV_{R_j} = \sum_{i=1}^{n} b_{ji} TV_{B_i} \left(\frac{1}{CR}\right)$$

where the units on a_{ij} and b_{ji} are $(\text{time})^{-1}$ and the units on $CB = CR = \sqrt{\lambda} = I$ are also $(\text{time})^{-1}$. Now regard $TV_{B_i} = \int_0^{\Delta T} X_{B_i} dt = X_{B_i} \Delta T$ where X_{B_i} is the total value added to type B_i per unit time. Here we assume that X_{B_i} is constant over a short time interval ΔT. Note X_{B_i} has units $(\text{time})^{-1}$.

The *short time* rate of contribution to value in the engagement, i.e. importance, is then proportional to the rate of contribution to opposing total value that the weapon eliminates:

$$X_{B_i} = \sum_j a_{ij} X_{R_j} \left(\frac{1}{CB} \right).$$

Now relax assumption that CB is constant, making it dependent on the i, j pair. This is a kind of disaggregation into separate interactions! Set "$CB_{i,j}$" $= \sqrt{X_{B_i} X_{R_j}}$ which has units $= (\text{time})^{-1}$, and view "$CB_{i,j}$" as a time constant for the i, j interaction. If we then set f_{ij} equal to the fractional involvement of B_i in kills of R_j, i.e. $f_{ij} = \frac{(\Delta R_j)_i}{\Delta R_j}$, and if we replace

$$\frac{1}{CB} \quad \text{with} \quad \frac{1}{\sqrt{X_{B_i} X_{R_j}}}$$

and reaggregate with a weighted sum of squares, then

$$X_{B_i} = \sqrt{\sum_j a_{ij}^2 X_{R_j}^2 f_{ij} / X_{B_i} X_{R_j}}$$

$$\text{or} \qquad X_{B_i}^3 = \sum_j a_{ij}^2 f_{ij} X_{R_j}.$$

These are exactly the ATCAL equations!

The assumption that $\sqrt{X_{B_i} X_{R_j}}$ is a time constant – in particular that $\frac{1}{\sqrt{X_{B_i} X_{R_j}}}$ is some kind of time to completion of the i, j-interaction is a weak one. For example, in a simple Lanchester model:

$$\frac{dB_1(t)}{dt} = -L_{11} R_1(t) \qquad L_{11} > 0$$

$$\frac{dR_1(t)}{dt} = -K_{11} B_1(t) \qquad K_{11} > 0$$

where at $t = 0$, $B_1(0) > 0$ and $R_1(0) > 0$, and at $t = t_f$, $B_1(t_f) > 0$ and $R_1(t_f) > 0$ with t_f is being duration of combat, we would have the intensity $I = \sqrt{K_{11} L_{11}}$ being the "time constant" with units $= (\text{time})^{-1}$.

Computations yield a time to completion of

$$t_f = \frac{\sinh^{-1}\left(\frac{\sqrt{1-a_0^2}\sqrt{1-b_0^2}}{a_0+b_0}\right)}{I}$$

where

$$a_0 = \frac{B_1(t_f)}{B_1(0)} < 1$$

$$b_0 = \frac{R_1(t_f)}{R_1(0)} < 1$$

are the fractions remaining at end of engagement. In short the time to completion is not $\frac{1}{I}$.

In addition, we recall that the eigenvalue method depends on constant kill rates to get non-time dependent values. Thus if our attrition model is not of Lanchester type:

$$\frac{dX}{dt} = MX \qquad M \text{ constant}$$

the values should either be time varying or thought of as instantaneous. On the other hand, ATCAL postulates the attrition as a function of time via its exponential averaging assumption, given initial and (derived) final forces. COSAGE does even better, giving actual time dependent loss data. This allows for the possibility of better (time dependent) values and (time dependent) shifting of of fires, and would move the ATCAL importances closer to our Fire Allocation Methods, which do exhibit shifting allocations of fire and changing values over time. As it stands, ATCAL gives no time dependent information; it is a static model giving the endpoint as a function of the initial point. In fact, ATCAL postulates an exponential attrition path between the initial and final points, while Lanchester postulates a quadratic attrition path.

Finally one can consider some extreme cases. For example, if we cause Blue weapons to kill twice as many Red in time ΔT then the ratio of importance doesn't double. ATCAL gives $\sqrt{2}$ as the relative factor. Another such case to consider is when many Blue types oppose one Red type:

$$X_1, \ldots, X_n \quad \text{Blue} \qquad\qquad X_0 \quad \text{Red}$$

Set $c = \sum_{k=1}^{n} a_{k0}$, then the ATCAL importances become:

$$X_0 = c^{-1/8} \left(\sum_{i=1}^{n} a_{0i}^2 a_{i0} \right)^{3/8}$$

$$X_i = c^{-3/8} a_{i0} \left(\sum_{i=1}^{n} a_{0i}^2 a_{i0} \right)^{1/8} = \frac{a_{i0}}{\sqrt[3]{c}} X_0^{1/3}$$

Now letting f_{i0} equal the fractional involvement of X_i in the kills of X_0, so that $f_{i0} = \frac{a_{i0}}{c}$, we get a relative value

$$\frac{X_i}{X_0} = X_0^{-2/3} c^{2/3} f_{i0}$$

which strangely depends on X_0.

§4. Numerical Comparisons and Sensitivity – Results of Some Simple Numerical Experiments.

We conducted a large number of simple numerical experiments for each of the three weapon scoring methods discussed above. Our purposes were two-fold. First, we wanted to examine how each method responded to various changes in its inputs. This led us to compute the amounts of change, certain numerical derivates, and the distributions obtained by treating various inputs as random variables. Our objective was to determine if the models responded in reasonable and consistent ways. Second, to the extent possible, we sought comparisons across methods that would allow us to evaluate the relative merits of each method.

On the whole, the results offered no surprises; and while the methods produced different values for various weapon systems, the results were usually qualitatively the same for all three methods. For that reason, we shall describe below only a small portion of the number of runs actually done. This sample is a good representative cross-section of the entire body of results.

Eigenvalue or Potential/Antipotential Method.

Following our notation in section one, we take as initial forces:

$$\begin{pmatrix} B_1 \\ B_2 \\ B_3 \end{pmatrix} = \begin{pmatrix} 50 \\ 24 \\ 15 \end{pmatrix}$$

$$\begin{pmatrix} R_1 \\ R_2 \\ R_3 \end{pmatrix} = \begin{pmatrix} 100 \\ 40 \\ 20 \end{pmatrix}$$

and attrition rates:

$$K = \begin{pmatrix} .1 & .05 & 0 \\ .075 & .1 & .025 \\ .025 & .025 & .075 \end{pmatrix}$$

$$L = \begin{pmatrix} .075 & .05 & .01 \\ .025 & .1 & .03 \\ .01 & .01 & .04 \end{pmatrix}.$$

Recall that K_{ij} is the rate at which one Blue weapon of type i ($i = 1, 2$, or 3) kills Red weapons of type j ($j = 1, 2$, or 3) and that L_{ji} is the rate at which a unit of Red j kills Blue i.

The eigenvalue method gives values:

$$\begin{pmatrix} VB_1 \\ VB_2 \\ VB_3 \end{pmatrix} = \begin{pmatrix} 1.0 \\ 1.25447 \\ .480896 \end{pmatrix}$$
$$\begin{pmatrix} VR_1 \\ VR_2 \\ VR_3 \end{pmatrix} = \begin{pmatrix} .950282 \\ 1.09924 \\ .278556 \end{pmatrix}$$

so that the total Blue value is 87.3208 and the total Red value is 144.569. This makes for a Red to Blue force ratio of 1.6556 to 1. Note that VB_1 has been normalized to be 1. Also the common eigenvalue λ is .022497 and the intensity $I = \sqrt{\lambda} = .14999$.

Our first numerical experiment involved computing the numerical partial derivatives of the values, the eigenvalue, and the intensity as functions of K_{ij} and L_{ji}. The results for L_{11}, L_{12}, L_{13}, L_{21} and L_{31} are shown below.

deriv[K,L,NB,NR,{1,1}]//Timing

```
Approximate Numerical Derivatives (5 point)
lambda: 0.0723231
Sqrt[lambda]: 0.241093
VB[2]: -2.4844
VB[3]: -1.13314
VR[1]: 4.2359
VR[2]: -3.64993
VR[3]: -0.915579
B: -76.6228
R: 259.281
step=0.00001
{0.77 Second, Null}
```

deriv[K,L,NB,NR,{1,2}]//Timing

```
Approximate Numerical Derivatives (5 point)
lambda: 0.0907275
Sqrt[lambda]: 0.302445
VB[2]: -3.11662
VB[3]: -1.4215
VR[1]: 5.31383
VR[2]: -4.57875
VR[3]: -1.14857
B: -96.1214
R: 325.261
step=0.00001
{0.66 Second, Null}
```

deriv[K,L,NB,NR,{1,3}]//Timing

```
Approximate Numerical Derivatives (5 point)
lambda: 0.0347799
Sqrt[lambda]: 0.115941
VB[2]: -1.19474
VB[3]: -0.544925
VR[1]: 2.03703
VR[2]: -1.75524
VR[3]: -0.440299
B: -36.8476
R: 124.687
step=1. 10^-6
{0.77 Second, Null}
```

deriv[K,L,NB,NR,{2,1}]//Timing

```
Approximate Numerical Derivatives (5 point)
lambda: 0.0677829
Sqrt[lambda]: 0.225958
VB[2]: 1.77313
VB[3]: 0.0206782
VR[1]: -0.839129
VR[2]: 6.19742
VR[3]: -0.295911
B: 42.8652
R: 158.066
step=0.00001
{0.71 Second, Null}
```

deriv[K,L,NB,NR,{3,1}]//Timing

```
Approximate Numerical Derivatives (5 point)
lambda: 0.0242435
Sqrt[lambda]: 0.0808172
VB[2]: 1.47834
VB[3]: 3.78407
VR[1]: 0.233072
VR[2]: 1.1502
VR[3]: 7.62474
B: 92.2411
R: 221.81
step=1.  10^-6
{0.65 Second, Null}
```

For example increasing L_{12} amounts to making Red 1 more effective against Blue 2. As a result the value of Red 1, VR_1, increases, as does the eigenvalue λ, the intensity $I = \sqrt{\lambda}$, and the total Red value. All other values, including the total Blue value, decrease. At first this might seem odd, but the reader is urged to keep in mind that these values are relative values (relative to B_1), not absolute values, and this complicates the situation.

Recall also that $L_{12} = .05$ so that a 10% increase in effectiveness of Red 1 against Blue 2 amounts to an increase of .005 in L_{12}. This decreases the value of a unit of Blue 2 relative to a unit of Blue 1 by 3.11662 times .005 or .0155831. This is a relatively small change.

Next, we did a direct check on the numerical derivative by altering (increasing) the K_{ij} and L_{ji} entries by 1%, 5% and 10% and recomputing all quantities. Sample results appear below:

compentry[K,L,NB,NR,{1,1,}]

	Actual	+1%	+5%	+10%
lambda	0.022497	0.022551	0.02277	0.023045
VB[2]	1.2545	1.2526	1.2453	1.2363
VB[3]	0.4809	0.48005	0.4767	0.47262
VR[1]	0.95028	0.95346	0.9661	0.9818
VR[2]	1.0992	1.0965	1.0857	1.0725
VR[3]	0.27856	0.27787	0.27516	0.27185
B	87.321	87.264	87.037	86.761
R	144.57	144.76	145.54	146.52

compentry[K,L,NB,NR,{1,2}]

	Actual	+1%	+5%	+10%
lambda	0.022497	0.022542	0.022723	0.022949
VB[2]	1.2545	1.2529	1.2468	1.2394
VB[3]	0.4809	0.48019	0.4774	0.47403
VR[1]	0.95028	0.95293	0.96344	0.97636
VR[2]	1.0992	1.097	1.088	1.0771
VR[3]	0.27856	0.27798	0.27573	0.27299
B	87.321	87.273	87.085	86.856
R	144.57	144.73	145.38	146.18

compentry[K,L,NB,NR,{2,2}]

	Actual	+1%	+5%	+10%
lambda	0.022497	0.022582	0.022924	0.023353
VB[2]	1.2545	1.2567	1.2655	1.2762
VB[3]	0.4809	0.48092	0.48103	0.48116
VR[1]	0.95028	0.94923	0.94504	0.93983
VR[2]	1.0992	1.107	1.138	1.1767
VR[3]	0.27856	0.27819	0.27671	0.27489
B	87.321	87.374	87.587	87.847
R	144.57	144.77	145.56	146.55

comprow[K,L,NB,NR,1]

	Actual	+1%	+5%	+10%
lambda	0.022497	0.0226	0.023014	0.023534
VB[2]	1.2545	1.251	1.2373	1.2212
VB[3]	0.4809	0.47929	0.47307	0.46572
VR[1]	0.95028	0.9563	0.98005	1.009
VR[2]	1.0992	1.0941	1.074	1.0501
VR[3]	0.27856	0.27726	0.27222	0.26623
B	87.321	87.212	86.792	86.295
R	144.57	144.94	146.41	148.23

compcol[K,L,NB,NR,1]

	Actual	+1%	+5%	+10%
lambda	0.022497	0.022571	0.022866	0.023238
VB[2]	1.2545	1.2532	1.2482	1.2422
VB[3]	0.4809	0.48043	0.47861	0.4764
VR[1]	0.95028	0.95327	0.96516	0.97989
VR[2]	1.0992	1.0982	1.094	1.089
VR[3]	0.27856	0.27856	0.27859	0.27865
B	87.321	87.283	87.137	86.959
R	144.57	144.82	145.85	147.12

Note that comprow[K,L,NB,NR,1] changes row 1 of L increasing L_{11}, L_{12}, L_{13} all by the same percentage. This corresponds to making Red 1 more effective against all Blue types. Similarly compcol[K,L,NB,NR,1] changes column 1 of L increasing L_{11}, L_{21}, L_{31}; in effect making all Red weapons more effective against Blue 1.

Finally, we treated various entries K_{ij} and L_{ji} as random variables. Besides varying them one at a time, we varied whole columns and rows of K and L by varying the entries both independently and together (correlated). We also treated the entire matrix K and L as random. We used both normal distributions and beta distributions to model the stochastic behavior.

Typical results are shown below. In this first example L_{11} is treated as a random variable subject to a normal distribution with mean .075 (the original value of L_{11}) and standard deviation equal to 5% of the mean value, i.e. .00375. The program did 1500 draws from this distribution and calculated the distributions for the Red and Blue values.

normentry[K,L,NB,NR,1500,.05,{1,1}]//Timing

```
Lambda: 0.022497
Mean: 0.0225001
Median: 0.0225026
Variance: 7.07888 10^-8
Standard Deviation: 0.000266062
Range: 0.00163993
Minimum: 0.0216807
Maximum: 0.0233206

Sqrt[Lambda]: 0.14999
Mean: 0.149998
Median: 0.150009
Variance: 7.86735 10^-7
Standard Deviation: 0.000886981
Range: 0.00546728
Minimum: 0.147244
Maximum: 0.152711

VB[2]: 1.25447
Mean: 1.25453
Median: 1.25428
Variance: 0.0000837406
Standard Deviation: 0.00915099
Range: 0.0564821
Minimum: 1.22768
Maximum: 1.28417
```

VB[3]: 0.480896
Mean: 0.480924
Median: 0.480809
Variance: 0.0000174203
Standard Deviation: 0.00417377
Range: 0.0257613
Minimum: 0.468676
Maximum: 0.494437

VR[1]: 1.
Mean: 0.950323
Median: 0.950607
Variance: 0.000243054
Standard Deviation: 0.0155902
Range: 0.0961508
Minimum: 0.901104
Maximum: 0.997255

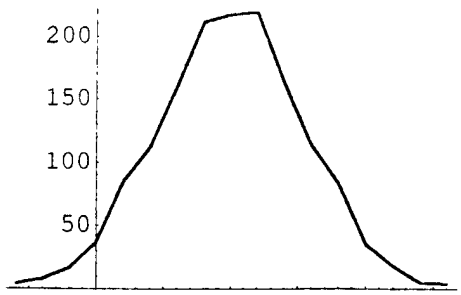
VR[2]: 1.25447
Mean: 1.09931
Median: 1.09896
Variance: 0.00018069
Standard Deviation: 0.0134421
Range: 0.0829561
Minimum: 1.05971
Maximum: 1.14266

VR[3]: 0.480896
Mean: 0.278573
Median: 0.278486
Variance: 0.0000113697
Standard Deviation: 0.0033719
Range: 0.020809
Minimum: 0.268638
Maximum: 0.289447

B: 87.3208
Mean: 87.3227
Median: 87.3149
Variance: 0.0796539
Standard Deviation: 0.28223
Range: 1.74199
Minimum: 86.4946
Maximum: 88.2366

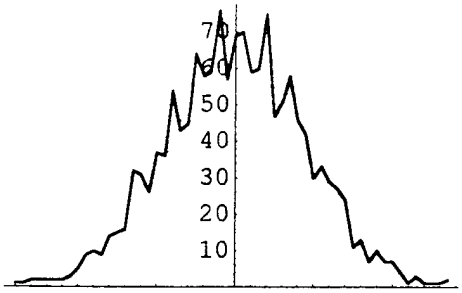
R: 144.569
Mean: 144.576
Median: 144.589
Variance: 0.910026
Standard Deviation: 0.953953
Range: 5.88066
Minimum: 141.606
Maximum: 147.487

Lambda Distribution



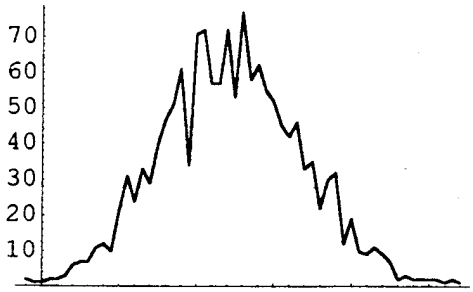
0.021705 022 0222502250227505 023 02325

Sqrt[Lambda] Distribution



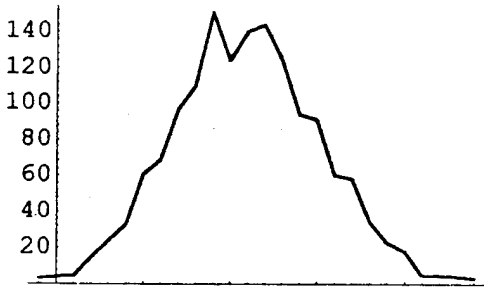
0.1480.149 0.15 0.1510.152

VB[2] Distribution



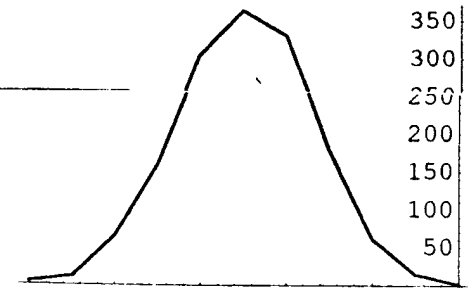
1.23 1.24 1.25 1.26 1.27 1.28

VB[3] Distribution



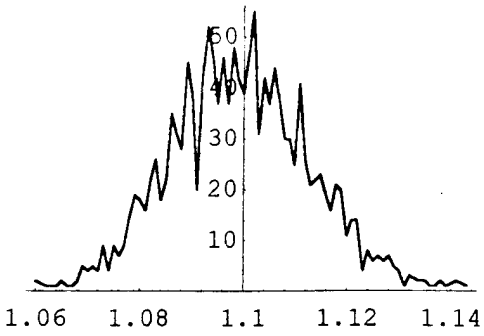
0.47 0.475 0.48 0.485 0.49

VR[1] Distribution

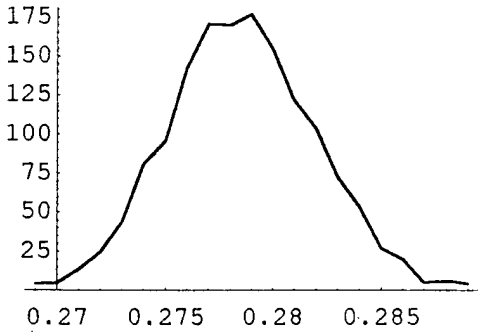


0.9 0.92 0.94 0.96 0.98 1

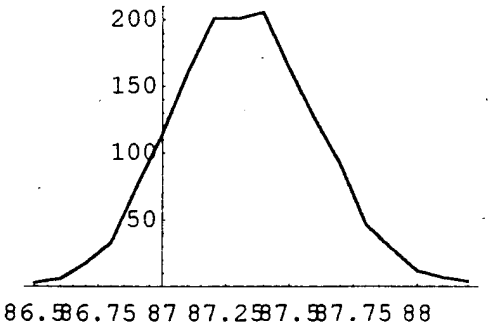
VR[2] Distribution



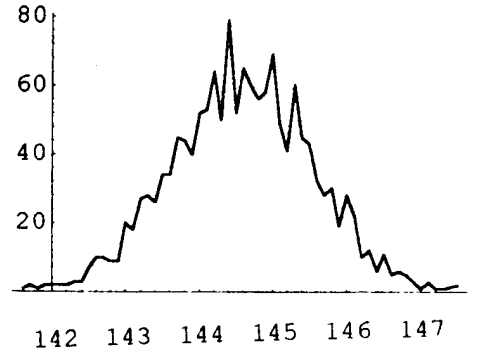
VR[3] Distribution



B Distribution



R Distribution



In order to make comparisons with ATCAL, it was necessary to create examples where the killer-victim scoreboard yielded essentially the same kill rates as those used in the eigenvalue examples. To some extent this is a guess, in that the attrition model is unspecified. We simply assumed straight line attrition.

For example, take initial forces of

$$\begin{pmatrix} B_1 \\ B_2 \\ B_3 \end{pmatrix} = \begin{pmatrix} 2000 \\ 960 \\ 600 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} R_1 \\ R_2 \\ R_3 \end{pmatrix} = \begin{pmatrix} 4000 \\ 1600 \\ 800 \end{pmatrix}$$

with killer victim scoreboard

		victim					
		B_1	B_2	B_3	R_1	R_2	R_3
	B_1	0	0	0	200	100	0
	B_2	0	0	0	72	96	24
killer	B_3	0	0	0	15	15	45
	R_1	300	200	40	0	0	0
	R_2	40	160	48	0	0	0
	R_3	8	8	32	0	0	0

Assuming a duration of 1 time unit and straight line attrition, we get

$$K = \begin{pmatrix} .1 & .05 & 0 \\ .075 & .1 & .025 \\ .025 & .025 & .075 \end{pmatrix} \qquad L = \begin{pmatrix} .075 & .05 & .01 \\ .025 & .01 & .03 \\ .01 & .01 & .04 \end{pmatrix}$$

as we had in the previous eigenvalue method examples. Using ATCAL, we obtain relative importances of

$$VB_1 = 1 \qquad VR_1 = .94595$$

$$VB_2 = 1.5513 \qquad VR_2 = 1.5396$$

$$VB_3 = 1.7971 \qquad VR_3 = 1.0477$$

This result appears to differ from the eigenvalue result in that both Red and Blue type 3 appear more valuable.

The numerical derivatives with respect to varying various L_{ji} (or equivalently varying entries in the killer-victim scoreboard) are shown below. We give both the absolute derivatives and the relative derivatives with VB_1 normalized to 1.

```
BK={{200,100,0},{72,96,24},{15,15,45}};
RK={{300,200,40},{40,160,48},{8,8,32}};
NB={2000,960,600};
NR={4000,1600,800};
```

```
deriv[BK,RK,NB,NR,{1,1}]//Timing
```

```
Approximate Numerical Derivatives (5 point)
VB[1]: 0.0000373978
VB[2]: 6.85268 10^-6
VB[3]: 1.31578 10^-7
VR[1]: 0.000179095
VR[2]: 1.07447 10^-6
VR[3]: -9.29081 10^-8
{9.5 Second, Null}
```

```
deriv[BK,RK,NB,NR,{1,2}]//Timing
```

```
Approximate Numerical Derivatives (5 point)
VB[1]: 0.0000393462
VB[2]: -0.0000469537
VB[3]: -6.93696 10^-7
VR[1]: 0.000270991
VR[2]: -0.000193113
VR[3]: -5.39146 10^-7
{9.23 Second, Null}
```

```
deriv[BK,RK,NB,NR,{1,3,}]//Timing
```

```
Approximate Numerical Derivatives (5 point)
VB[1]: 3.38649 10^-6
VB[2]: -0.0000430005
VB[3]: -0.000275548
VR[1]: 0.0000618685
VR[2]: -0.000107301
VR[3]: -0.00048685
{9.45 Second, Null}
```

```
deriv[BK,RK,NB,NR{2,1}]//Timing
```

```
Approximate Numerical Derivatives (5 point)
VB[1]: -5.0077 10^-6
VB[2]: 1.99616 10^-6
VB[3]: -2.68284 10^-8
VR[1]: -0.0000284272
VR[2]: 0.0000103152
VR[3]: -5.92568 10^-8
{11.09 Second, Null}
```

deriv[BK,RK,NB,NR,{3,1}]//Timing

```
Approximate Numerical Derivatives (5 point)
VB[1]: -3.02879 10^-6
VB[2]: -4.93462 10^-7
VB[3]: 2.19239 10^-6
VR[1]: -0.0000144314
VR[2]: -2.592 10^-7
VR[3]: 3.90199 10^-6
{13.34 Second, Null}
```

deriv[BK,RK,NB,NR,{2,2}]//Timing

```
Approximate Numerical Derivatives (5 point)
VB[1]: 0.000040094
VB[2]: 0.000136194
VB[3]: 1.74046 10^-6
VR[1]: -4.4331 10^-6
VR[2]: 0.000463299
VR[3]: 2.78241 10^-7
{12.42 Second, Null}
```

BK={{200,100,0},{72,96,24},{15,15,45}};
RK={{300,200,40},{40,160,48},{8,8,32}};
NB={2000,960,600};
NR={4000,1600,800};

deriv[BK,RK,NB,NR,{1,1,}]//Timing

```
Approximate Numerical Derivatives (5 point)
VB[2]: -1.37032
VB[3]: -1.79654
VR[1]: 3.84922
VR[2]: -1.51334
VR[3]: -1.05191
{12.14 Second, Null}
```

deriv[BK,RK,NB,NR,{1,2}]//Timing

```
Approximate Numerical Derivatives (5 point)
VB[2]: -2.89237
VB[3]: -1.91243
VR[1]: 6.2611
VR[2]: -6.79461
VR[3]: -1.11854
{13.78 Second, Null}
```

deriv[BK,RK,NB,NR,{1,3}]//Timing

```
Approximate Numerical Derivatives (5 point)
VB[2]: -1.29239
VB[3]: -7.54303
VR[1]: 1.57123
VR[2]: -3.0135
VR[3]: -13.1236
{20.55 Second, Null}
```

deriv[BK,RK,NB,NR,{2,1}]//Timing

```
Approximate Numerical Derivatives (5 point)
VB[2]: 0.26153
VB[3]: 0.240317
VR[1]: -0.634494
VR[2]: 0.482767
VR[3]: 0.138934
{9.45 Second, Null}
```

deriv[BK,RK,NB,NR,{3,1}]//Timing

```
Approximate Numerical Derivatives (5 point)
VB[2]: 0.112628
VB[3]: 0.204504
VR[1]: -0.309782
VR[2]: 0.117952
VR[3]: 0.189498
{9.44 Second, Null}
```

deriv[BK,RK,NB,NR{2,2}]//Timing

```
Approximate Numerical Derivatives (5 point)
VB[2]: 1.98182
VB[3]: -1.88323
VR[1]: -1.13453
VR[2]: 10.7553
VR[3]: -1.11763
{9.4 Second, Null}
```

For the Fire Allocation method, we follow the simple model as outlined in Section 2.a.ii. NB and NR represent the initial forces and A and C are the pure kill rates following the notation of Section 2.a.ii, except that the indices on a_{ij} and c_{ij} have been switched. Thus a_{21} in the program is a_{12} in the text of Section 2.a.ii. Note also that the value h_1 is normalized to 1.

Allocation Method
Base Program
June 3, 1994

This program determines the weapon values given the matrices A and C and the strength of the forces NB and NR.

```
Off[Syntax::newl,General::spell,General::spell1,Part::partd]
allocate[{{a11_,a12_},{a21_,a22_}},{{c11_,c12_},{c21_,c22_}},
    {b01_,b02_},{r01_,r02_}]:=

Block[{list1,list2,list3,list4,phi,psi,matx,philist,psilist,t1,
    t2,i,j,common,lam,x,list,xvec},

list1={-a11 b01^2 c11^2 r01+b01^2 c11 phi r01-a12 b01^2 c21^2 r02
    +b01^2 c21 phi r02, b01 b02 c11 phi r01 + b01 b02 c21 phi
    r02, -2 a11 b01 b02 c11 c12 r01 + b01 b02 c12 phi r01 - 2 a12
    b01 b02 c21 c22 r02 + b01 b02 c22 phi r02, b02^2 c12 phi r01
    + b02^2 c22 phi r02, -a11 b02^2 c12^2 r01 - a12 b02^2 c22^2
    r02,0,0,0};
matx={};
For[i=0,i<=3,i++,AppendTo[matx,RotateRight[list1,i]]];
list2={-a21 b01^2 c11^2 r01 - a22 b01^2 c21^2 r02,
    b01^2 c11 phi r01 + b01^2 c21 phi r02, -2 a21 b01 b02 c11 c12
    r01 + b01 b02 c11 phi r01 - 2 a22 b01 b02 c21 c22 r02 + b01
    b02 c21 phi r02, b01 b02 c12 phi r01 + b01 b02 c22 phi r02,
    -a21 b02^2 c12^2 r01 + b02^2 c12 phi r01 - a22 b02^2 c22^2
    r02 + b02^2 c22 phi r02,0,0,0};
For[i=0,i<=3,i++,AppendTo[matx,RotateRight[list2,i]]];
philist=Select[phi /. NSolve[Det[matx]==0,phi],Im[#]==0 &];
list3={-a11^2 b01 c11 r01^2 - a21^2 b02 c12 r01^2 + a11 b01
    psi r01^2 + a21 b02 psi r01^2, a11 b01 psi r01 r02 +
    a21 b02 psi r01 r02, -2 a11 a12 b01 c11 r01 r02 - 2 a21 a22
    b02 c12 r01 r02 + a12 b01 psi r01 r02 + a22 b02 psi r01 r02,
    a12 b01 psi r02^2 + a22 b02 psi r02^2, -a12^2 b01 c11 r02^2 -
    a22^2 b02 c12 r02^2,0,0,0};
matx={};
For[i=0,i<=3,i++,AppendTo[matx,RotateRight[list3,i]]];
list4={-a11^2 b01 c21 r01^2 - a21^2 b02 c22 r01^2, a11 b01 psi
    r01^2 + a21 b02 psi r01^2, -2 a11 a12 b01 c21 r01 r02 - 2 a21
    a22 b02 c22 r01 r02 + a11 b01 psi r01 r02 + a21 b02 psi r01
    r02, a12 b01 psi r01 r02 + a22 b02 psi r01 r02, -a12^2 b01
    c21 r02^2 - a22^2 b02 c22 r02^2 + a12 b01 psi r02^2 + a22 b02
    psi r02^2,0,0,0};
For[i=0,i<=3,i++,AppendTo[matx,RotateRight[list4,i]]];
psilist=Select[psi /. NSolve[Det[matx]==0,psi],Im[#]==0 &];

common={};
t1=Length[philist];
t2=Length[psilist];
For[i=1,i<=t1,i++,
```

```
    For[j=1,j<=t2,j++,
        If[ Abs[philist[[i]]-psilist[[j]] ]<10^-10,
            AppendTo[common,philist[[i]] ]
        ];
    ];
];
If[Length[common]==0,Break[] ];
lam=Max[common];
Print["Common Lambda Value: ",lam];

xvec={x^4,x^3,x^2,x,1};
list=Take[list1,5] /. phi->lam;
xvals1=Select[x /. Solve[list.xvec==0,x],Im[#]==0 &];
list=Take[list2,5] /. phi->lam;
xvals2=Select[x /. Solve[list.xvec==0,x],Im[#]==0 &];
common={};
t1=Length[xvals1];
t2=Length[xvals2];
For[i=1,i<=t1,i++,
    For[j=1,j<=t2,j++,
        If[ Abs[xvals1[[i]]-xvals2[[j]] ]<10^-10,
            AppendTo[common,xvals1[[i]] ]
        ];
    ];
];
If[Length[common]==0,Break[] ];
ratio=Max[common];

h1=1;
h2=1/ratio;
g1=(b01 c11 h1^2 + b02 c12 h2^2)/(Sqrt[lam]*(b01 h1 + b02 h2));
g2=(b01 c21 h1^2 + b02 c22 h2^2)/(Sqrt[lam]*(b01 h1 + b02 h2));

Print["H[1]=",h1];
Print["H[2]=",h2];
Print["G[1]=",g1];
Print["G[2]=",g2]
]
allocate[{{.1,.05},{.075,.1}},{{.075,.05},{.025,.1}},{50,25},
    {100,40}]
```

```
Common Lambda Value: 0.00561786
H[1]=1
H[2]=0.849138
G[1]=0.871234
G[2]=0.571781
```

Allocation Method
Numerical Derivatives
June 3, 1994

This program accepts the the input data of the A and C matrices, the strength of the forces NB and NR, and the indices of an element in the C matrix, returning the numerical derivatives of the weapon values with respect to this element.

```
Off[Syntax::newl,General::spell,General::spell1,Part::partd]

deriv[A_,C_,NB_,NR_,ind_]:=
Block[{NA,NC,step,Rlist,Blist,base,process,b01,b02,r01,r02,x,
       xvec,vals},

base[matA_,matC_]:=
Block[{a11,a12,a21,a22,c11,c12,c21,c22,list1,list2,list3,list4,
       matx,i,j,phi,philist,psi,psilist,common,t1,t2,lam,list,
       xvals1,xvals2,ratio,g1,g2,h2},

{{a11,a12},{a21,a22}}=matA;
{{c11,c12},{c21,c22}}=matC;

matx={};
list1={-a11 b01^2 c11^2 r01+b01^2 c11 phi r01-a12 b01^2 c21^2 r02
    +b01^2 c21 phi r02, b01 b02 c11 phi r01 + b01 b02 c21 phi
    r02, -2 a11 b01 b02 c11 c12 r01 + b01 b02 c12 phi r01 - 2 a12
    b01 b02 c21 c22 r02 + b01 b02 c22 phi r02, b02^2 c12 phi r01
    + b02^2 c22 phi r02, -a11 b02^2 c12^2 r01 - a12 b02^2 c22^2
    r02,0,0,0};
For[i=0,i<=3,i++,AppendTo[matx,RotateRight[list1,i]]];
list2={-a21 b01^2 c11^2 r01 - a22 b01^2 c21^2 r02,
    b01^2 c11 phi r01 + b01^2 c21 phi r02, -2 a21 b01 b02 c11 c12
    r01 + b01 b02 c11 phi r01 - 2 a22 b01 b02 c21 c22 r02 + b01
    b02 c21 phi r02, b01 b02 c12 phi r01 + b01 b02 c22 phi r02,
    -a21 b02^2 c12^2 r01 + b02^2 c12 phi r01 - a22 b02^2 c22^2
    r02 + b02^2 c22 phi r02,0,0,0};
For[i=0,i<=3,i++,AppendTo[matx,RotateRight[list2,i]]];
philist=Select[phi /. NSolve[Det[matx]==0,phi],Im[#]==0 &];

matx={};
list3={-a11^2 b01 c11 r01^2 - a21^2 b02 c12 r01^2 + a11 b01
    psi r01^2 + a21 b02 psi r01^2, a11 b01 psi r01 r02 +
    a21 b02 psi r01 r02, -2 a11 a12 b01 c11 r01 r02 - 2 a21 a22
    b02 c12 r01 r02 + a12 b01 psi r01 r02 + a22 b02 psi r01 r02,
    a12 b01 psi r02^2 + a22 b02 psi r02^2, -a12^2 b01 c11 r02^2 -
    a22^2 b02 c12 r02^2,0,0,0};
For[i=0,i<=3,i++,AppendTo[matx,RotateRight[list3,i]]];
list4={-a11^2 b01 c21 r01^2 - a21^2 b02 c22 r01^2, a11 b01 psi
    r01^2 + a21 b02 psi r01^2, -2 a11 a12 b01 c21 r01 r02 - 2 a21
    a22 b02 c22 r01 r02 + a11 b01 psi r01 r02 + a21 b02 psi r01
    r02, a12 b01 psi r01 r02 + a22 b02 psi r01 r02, -a12^2 b01
    c21 r02^2 - a22^2 b02 c22 r02^2 + a12 b01 psi r02^2 + a22 b02
```

```
    psi r02^2,0,0,0};
For[i=0,i<=3,i++,AppendTo[matx,RotateRight[list4,i]]];
psilist=Select[psi /. NSolve[Det[matx]==0,psi],Im[#]==0 &];

common={};
t1=Length[philist];
t2=Length[psilist];
For[i=1,i<=t1,i++,
    For[j=1,j<=t2,j++,
        If[ Abs[philist[[i]]-psilist[[j]] ]<10^-10,
            AppendTo[common,philist[[i]] ]
        ];
    ];
];

If[Length[common]==0,
    Print["*** NO COMMON LAMBDA VALUE! ***"];
    Break[]
];
If[Length[common]>1,
    Print["*** MULTIPLE COMMON LAMBDA VALUES! ***"]
];
lam=Max[common];

list=Take[list1,5] /. phi->lam;
xvals1=Select[x /. Solve[list.xvec==0,x],Im[#]==0 &];
list=Take[list2,5] /. phi->lam;
xvals2=Select[x /. Solve[list.xvec==0,x],Im[#]==0 &];
common={};
t1=Length[xvals1];
t2=Length[xvals2];
For[i=1,i<=t1,i++,
    For[j=1,j<=t2,j++,
        If[ Abs[xvals1[[i]]-xvals2[[j]] ]<10^-10,
            AppendTo[common,xvals1[[i]] ]
        ];
    ];
];
If[Length[common]==0,
    Print["*** NO COMMON RATIO! ***"];
    Break[]
];
If[Length[common]>1,
    Print["*** MULTIPLE COMMON RATIOS! ***"]
];
ratio=Max[common];

h2=1/ratio;
g1=(b01 c11 + b02 c12 h2^2)/(Sqrt[lam]*(b01 + b02 h2));
g2=(b01 c21 + b02 c22 h2^2)/(Sqrt[lam]*(b01 + b02 h2));
{{g1,g2},h2}
```

```
];

process[vals_]:=
Block[{},
    AppendTo[Blist,vals[[2]] ];
    AppendTo[Rlist[[1]],vals[[1,1]] ];
    AppendTo[Rlist[[2]],vals[[1,2]] ];
];

calc[x_]:=(x[[1]]-8x[[2]]+8x[[3]]-x[[4]])/(12 step);

    NA=A;
    NC=C;
    step=1;
    While[NC[[ind[[1]],ind[[2]] ]]-2*step<=0,step*=.1];
    step*=.001;
    {b01,b02}=NB;
    {r01,r02}=NR;
    xvec={x^4,x^3,x^2,x,1};
    Rlist={{},{}};
    Blist={};
    NC[[ind[[1]],ind[[2]] ]]-=2*step;
    process[base[NA,NC]];
    NC[[ind[[1]],ind[[2]] ]]+=step;
    process[base[NA,NC]];
    NC[[ind[[1]],ind[[2]] ]]+=2*step;
    process[base[NA,NC]];
    NC[[ind[[1]],ind[[2]] ]]+=step;
    process[base[NA,NC]];
    Print["Approximate Numerical Derivatives"];
    Print["G[1]: ",calc[Rlist[[1]]] ];
    Print["G[2]: ",calc[Rlist[[2]]] ];
    Print["H[2]: ",calc[Blist] ];
]
A={{.1,.05},{.075,.1}};
Cx={{.075,.05},{.025,.1}};
NB={50,25};
NR={100,40};
```

Derivatives with respect to C[1,1]

```
deriv[A,Cx,NB,NR,{1,1}]
```

```
Approximate Numerical Derivatives
G[1]: 4.29571
G[2]: -4.59138
H[2]: -2.36597
```

Derivatives with respect to C[1,2]

```
deriv[A,Cx,NB,NR,{1,2}]
```

```
Approximate Numerical Derivatives
G[1]: 1.54868
G[2]: -1.65528
H[2]: -0.852976
```

Derivatives with respect to C[2,1]

```
deriv[A,Cx,NB,NR,{2,1}]
```

```
Approximate Numerical Derivatives
G[1]: 0.589737
G[2]: 11.6787
H[2]: 3.60508
```

Derivatives with respect to C[2,2]

```
deriv[A,Cx,NB,NR,{2,2}]
```

```
Approximate Numerical Derivatives
G[1]: 0.212611
G[2]: 4.2104
H[2]: 1.2997
```

Allocation Method
Entry Sensitivity Analysis
May 27, 1994

This program receives the matrices A and C, the strength of the forces NB and NR, the number of iterations, the standard deviation of the distribution, and the indices of the entry in the C matrix to be altered. First, the program calculates the weapon values for the initial A and C. The program then repeatedly perturbs the indicated entry by a random draw from the given normal distribution. The statistics of the calculated weapon values are returned and the distributions are plotted.

```
Off[General::spell1,General::spell,Syntax::newl,Part::partd]
<<Statistics`DescriptiveStatistics`
<<Statistics`ContinuousDistributions`

normentry[A_,C_,NB_,NR_,iter_,stdev_,ind_]:=
Block[{NA,NC,Blist,Rlist,B,R,b01,b02,r01,r02,base,process,x,xvec,
       ndist,results,i},

base[matA_,matC_]:=
Block[{a11,a12,a21,a22,c11,c12,c21,c22,list1,list2,list3,list4,
       matx,i,j,phi,philist,psi,psilist,common,t1,t2,lam,list,
       xvals1,xvals2,ratio,g1,g2,h2},

{{a11,a12},{a21,a22}}=matA;
{{c11,c12},{c21,c22}}=matC;

matx={};
list1={-a11 b01^2 c11^2 r01+b01^2 c11 phi r01-a12 b01^2 c21^2 r02
    +b01^2 c21 phi r02, b01 b02 c11 phi r01 + b01 b02 c21 phi
    r02, -2 a11 b01 b02 c11 c12 r01 + b01 b02 c12 phi r01 - 2 a12
    b01 b02 c21 c22 r02 + b01 b02 c22 phi r02, b02^2 c12 phi r01
    + b02^2 c22 phi r02, -a11 b02^2 c12^2 r01 - a12 b02^2 c22^2
    r02,0,0,0};
For[i=0,i<=3,i++,AppendTo[matx,RotateRight[list1,i]]];
list2={-a21 b01^2 c11^2 r01 - a22 b01^2 c21^2 r02,
    b01^2 c11 phi r01 + b01^2 c21 phi r02, -2 a21 b01 b02 c11 c12
    r01 + b01 b02 c11 phi r01 - 2 a22 b01 b02 c21 c22 r02 + b01
    b02 c21 phi r02, b01 b02 c12 phi r01 + b01 b02 c22 phi r02,
    -a21 b02^2 c12^2 r01 + b02^2 c12 phi r01 - a22 b02^2 c22^2
    r02 + b02^2 c22 phi r02,0,0,0};
For[i=0,i<=3,i++,AppendTo[matx,RotateRight[list2,i]]];
philist=Select[phi /. NSolve[Det[matx]==0,phi],Im[#]==0 &];

matx={};
list3={-a11^2 b01 c11 r01^2 - a21^2 b02 c12 r01^2 + a11 b01
    psi r01^2 + a21 b02 psi r01^2, a11 b01 psi r01 r02 +
    a21 b02 psi r01 r02, -2 a11 a12 b01 c11 r01 r02 - 2 a21 a22
    b02 c12 r01 r02 + a12 b01 psi r01 r02 + a22 b02 psi r01 r02,
    a12 b01 psi r02^2 + a22 b02 psi r02^2, -a12^2 b01 c11 r02^2 -
    a22^2 b02 c12 r02^2,0,0,0};
For[i=0,i<=3,i++,AppendTo[matx,RotateRight[list3,i]]];
```

```
list4={-a11^2 b01 c21 r01^2 - a21^2 b02 c22 r01^2, a11 b01 psi
    r01^2 + a21 b02 psi r01^2, -2 a11 a12 b01 c21 r01 r02 - 2 a21
    a22 b02 c22 r01 r02 + a11 b01 psi r01 r02 + a21 b02 psi r01
    r02, a12 b01 psi r01 r02 + a22 b02 psi r01 r02, -a12^2 b01
    c21 r02^2 - a22^2 b02 c22 r02^2 + a12 b01 psi r02^2 + a22 b02
    psi r02^2,0,0,0};
For[i=0,i<=3,i++,AppendTo[matx,RotateRight[list4,i]]];
psilist=Select[psi /. NSolve[Det[matx]==0,psi],Im[#]==0 &];

common={};
t1=Length[philist];
t2=Length[psilist];
For[i=1,i<=t1,i++,
    For[j=1,j<=t2,j++,
        If[ Abs[philist[[i]]-psilist[[j]] ]<10^-10,
            AppendTo[common,philist[[i]] ]
        ];
    ];
];

If[Length[common]==0,
    Print["*** NO COMMON LAMBDA VALUE! ***"];
    Break[]
];
If[Length[common]>1,
    Print["*** MULTIPLE COMMON LAMBDA VALUES! ***"]
];
lam=Max[common];

list=Take[list1,5] /. phi->lam;
xvals1=Select[x /. Solve[list.xvec==0,x],Im[#]==0 &];
list=Take[list2,5] /. phi->lam;
xvals2=Select[x /. Solve[list.xvec==0,x],Im[#]==0 &];
common={};
t1=Length[xvals1];
t2=Length[xvals2];
For[i=1,i<=t1,i++,
    For[j=1,j<=t2,j++,
        If[ Abs[xvals1[[i]]-xvals2[[j]] ]<10^-10,
            AppendTo[common,xvals1[[i]] ]
        ];
    ];
];
If[Length[common]==0,
    Print["*** NO COMMON RATIO! ***"];
    Break[]
];
If[Length[common]>1,
    Print["*** MULTIPLE COMMON RATIOS! ***"]
];
ratio=Max[common];
```

```
h2=1/ratio;
g1=(b01 c11 + b02 c12 h2^2)/(Sqrt[lam]*(b01 + b02 h2));
g2=(b01 c21 + b02 c22 h2^2)/(Sqrt[lam]*(b01 + b02 h2));
{{g1,g2},h2}
];

process[v_,str_]:=
Block[{scale,ints,list,min,max,temp,i,P1},
    Print["Mean: ",Mean[v]];
    Print["Median: ",Median[v]];
    Print["Variance: ",Variance[v]];
    Print["Standard Deviation: ",StandardDeviation[v]];
    Print["Range: ",SampleRange[v]];
    Print["Minimum: ",min=Min[v]];
    Print["Maximum: ",max=Max[v]];
    scale=0;
    While[Round[max*10^scale]-Round[min*10^scale]<10,scale++];
    While[Round[max*10^scale]-Round[min*10^scale]>99,scale--];
    ints=Round[v*10^scale];
    list={};
    min=Min[ints];
    max=Max[ints];
    For[i=min,i<=max,i++,
        temp=N[{i/10^scale,Count[ints,i]}];
        If[temp[[2]]!=0,AppendTo[list,temp]]
    ];
    P1=ListPlot[list,PlotJoined->True,PlotLabel->str,
        PlotRange->{0,Automatic},Ticks->{{N[min/10^scale],
        N[(min+max)/(2*10^scale)],N[max/10^scale]},Automatic},
        AxesOrigin->{Min[v],0}]
];

NA=A; NC=C;
{b01,b02}=NB; {r01,r02}=NR;
xvec={x^4,x^3,x^2,x,1};
Blist=Rlist={};
{R,B}=base[NA,NC];

ndist=NormalDistribution[0,stdev*NC[[ind[[1]],ind[[2]] ]] ];
For[i=1,i<=iter,i++,
    NC=C;
    NC[[ind[[1]],ind[[2]] ]]+=Random[ndist];

    While[NC[[ind[[1]],ind[[2]] ]]<0 ||
        NC[[ind[[1]],ind[[2]] ]]>2*C[[ind[[1]],ind[[2]] ]],
        Print["*** ERROR - OUT OF BOUNDS ***"];

        Print[Abs[NC[[ind[[1]],ind[[2]] ]]
            -C[[ind[[1]],ind[[2]] ]] ] /
            (stdev*C[[ind[[1]],ind[[2]] ]]),"  stand. dev."];
```

```
        NC[[ind[[1]],ind[[2]] ]]=C[[ind[[1]],ind[[2]] ]]+
            Random[ndist];
    ];

    results=base[NA,NC];
    If[i==1,
        Rlist={{results[[1,1]]},{results[[1,2]]}};
        Blist={results[[2]]},
        AppendTo[Rlist[[1]],results[[1,1]] ];
        AppendTo[Rlist[[2]],results[[1,2]] ];
        AppendTo[Blist,results[[2]] ]
    ];
];

Print["H[2]: ",B];
process[Blist,"H[2] Distribution"];
Print[" "];
Print["G[1]: ",R[[1]] ];
process[Rlist[[1]],"G[1] Distribution"];
Print[" "];
Print["G[2]: ",R[[2]] ];
process[Rlist[[2]],"G[2] Distribution"];
]
```

Statistics from Sensitivity Analysis
Fire Allocation & Attrition Method
June 3, 1994

The following data was obtained by the sensitivity analysis routines for the Fire Allocation Method. The initial data for these routines were:

A={{.1,.05},{.075,.1}} NB={50,25}
C={{.075,.05},{.025,.1}} NR={100,40}

Statistics from perturbing the C[1,1] entry by a random draw from a normal distribution with 5% standard deviation.

```
H[2]: 0.849138
Mean: 0.850096
Median: 0.849214
Variance: 0.0000909999
Standard Deviation: 0.00953939
Range: 0.0576209
Minimum: 0.825638
Maximum: 0.883259

G[1]: 0.871234
Mean: 0.870513
Median: 0.871098
Variance: 0.000288957
Standard Deviation: 0.0169987
Range: 0.101812
Minimum: 0.819872
Maximum: 0.921684

G[2]: 0.571781
Mean: 0.573207
Median: 0.571927
Variance: 0.000336314
Standard Deviation: 0.0183389
Range: 0.110249
Minimum: 0.52304
Maximum: 0.633289
```

Statistics from simultaneously perturbing each entry in the first row of C by a random draw from a normal distribution with 5% standard deviation.

```
H[2]: 0.849138
Mean: 0.849598
Median: 0.848658
Variance: 0.000134497
Standard Deviation: 0.0115973
Range: 0.0714355
Minimum: 0.82365
Maximum: 0.895086

G[1]: 0.871234
Mean: 0.871896
Median: 0.872109
Variance: 0.000427619
Standard Deviation: 0.0206789
Range: 0.121111
Minimum: 0.805736
Maximum: 0.926846
```

```
G[2]: 0.571781
Mean: 0.572038
Median: 0.570848
Variance: 0.000496493
Standard Deviation: 0.0222821
Range: 0.134293
Minimum: 0.518575
Maximum: 0.652868
```

Statistics from independently perturbing each entry in the first row of C by a random draw from a normal distribution with 5% standard deviation.

```
H[2]: 0.849138
Mean: 0.849416
Median: 0.848763
Variance: 0.0000862001
Standard Deviation: 0.0092844
Range: 0.0625471
Minimum: 0.825587
Maximum: 0.888134

G[1]: 0.871234
Mean: 0.871701
Median: 0.871917
Variance: 0.000278728
Standard Deviation: 0.0166951
Range: 0.107946
Minimum: 0.813867
Maximum: 0.921813

G[2]: 0.571781
Mean: 0.571909
Median: 0.571052
Variance: 0.000320908
Standard Deviation: 0.0179139
Range: 0.118523
Minimum: 0.522927
Maximum: 0.64145
```

Statistics from simultaneously perturbing each entry in the first column of C by a random draw from a normal distribution with 5% standard deviation.

```
H[2]: 0.849138
Mean: 0.849248
Median: 0.849159
Variance: 0.0000196365
Standard Deviation: 0.00443131
Range: 0.0264521
Minimum: 0.839086
Maximum: 0.865538

G[1]: 0.871234
Mean: 0.871637
Median: 0.871154
Variance: 0.000286002
Standard Deviation: 0.0169116
Range: 0.0973707
Minimum: 0.817552
Maximum: 0.914923
```

G[2]: 0.571781
Mean: 0.572005
Median: 0.571793
Variance: 7.3137 10^{-6}
Standard Deviation: 0.00270439
Range: 0.0166747
Minimum: 0.566729
Maximum: 0.583403

Statistics from independently perturbing each entry in the first column of C by a random draw from a normal distribution with 5% standard deviation.

H[2]: 0.849138
Mean: 0.850224
Median: 0.849267
Variance: 0.000115784
Standard Deviation: 0.0107603
Range: 0.0654364
Minimum: 0.818917
Maximum: 0.884353

G[1]: 0.871234
Mean: 0.870498
Median: 0.870842
Variance: 0.000298903
Standard Deviation: 0.0172888
Range: 0.100964
Minimum: 0.816978
Maximum: 0.917942

G[2]: 0.571781
Mean: 0.573444
Median: 0.572479
Variance: 0.000563436
Standard Deviation: 0.0237368
Range: 0.146793
Minimum: 0.495818
Maximum: 0.642611

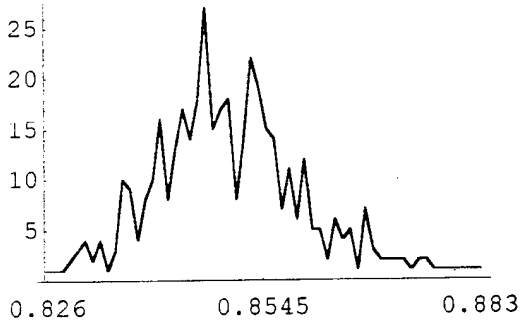
Graphs from Sensitivity Analysis
Fire Allocation & Attrition Method
June 3, 1994

The following graphs were obtained by the sensitivity analysis routines for the Fire Allocation Method. The initial data for these routines were:

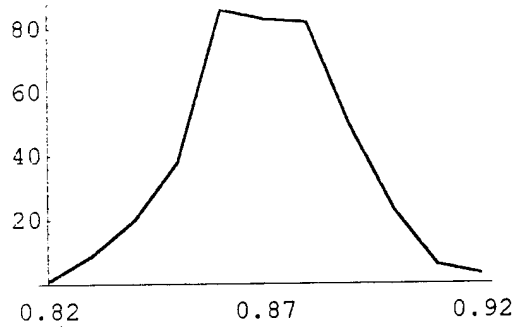
A={{.1,.05},{.075,.1}} NB={50,25}
C={{.075,.05},{.025,.1}} NR={100,40}

Graphs from perturbing the C[1,1] entry by a random draw from a normal distribution with 5% standard deviation.

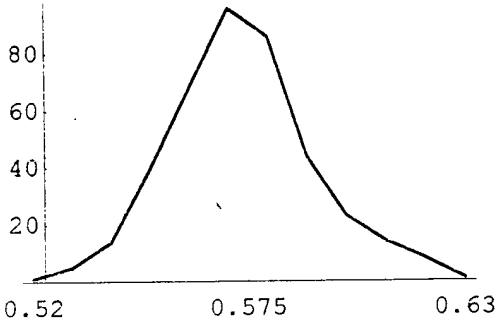
H[2] Distribution



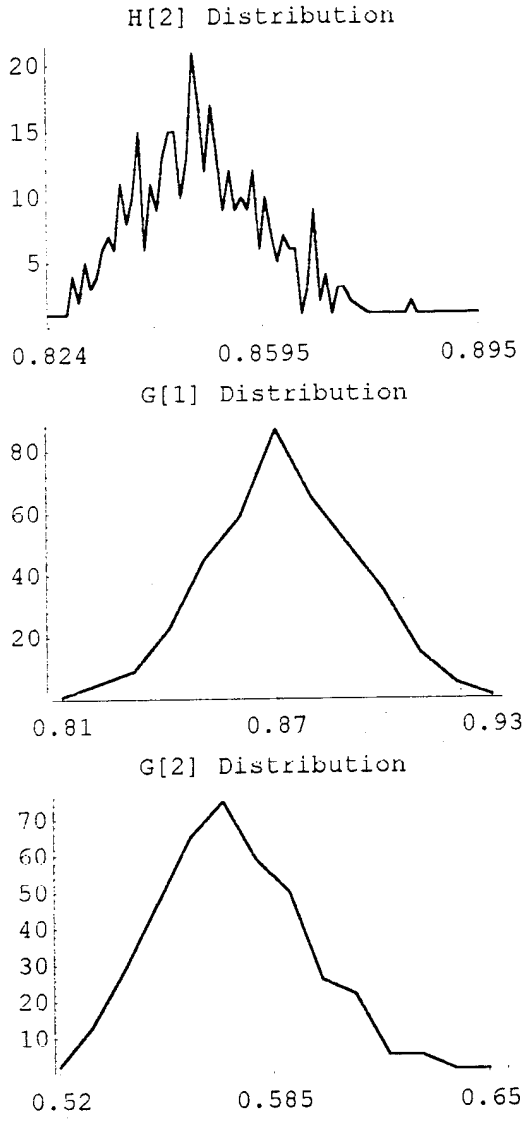
G[1] Distribution



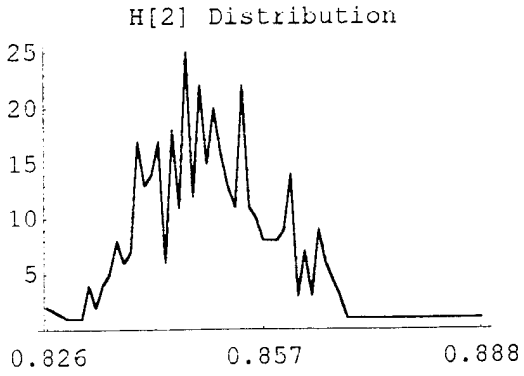
G[2] Distribution

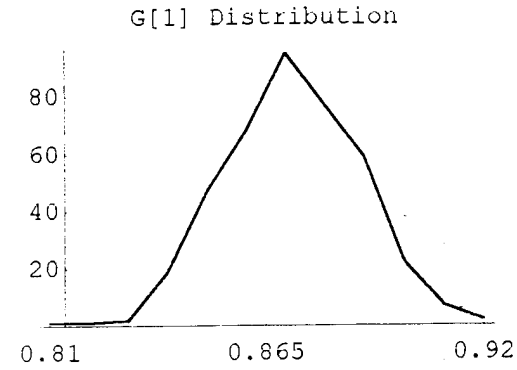


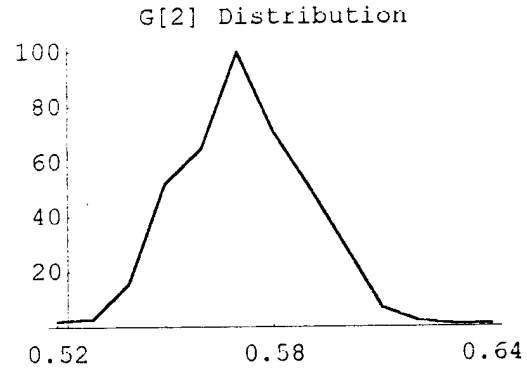
Graphs from simultaneously perturbing each entry in the first row of C by a random draw from a normal distribution with 5% standard deviation.



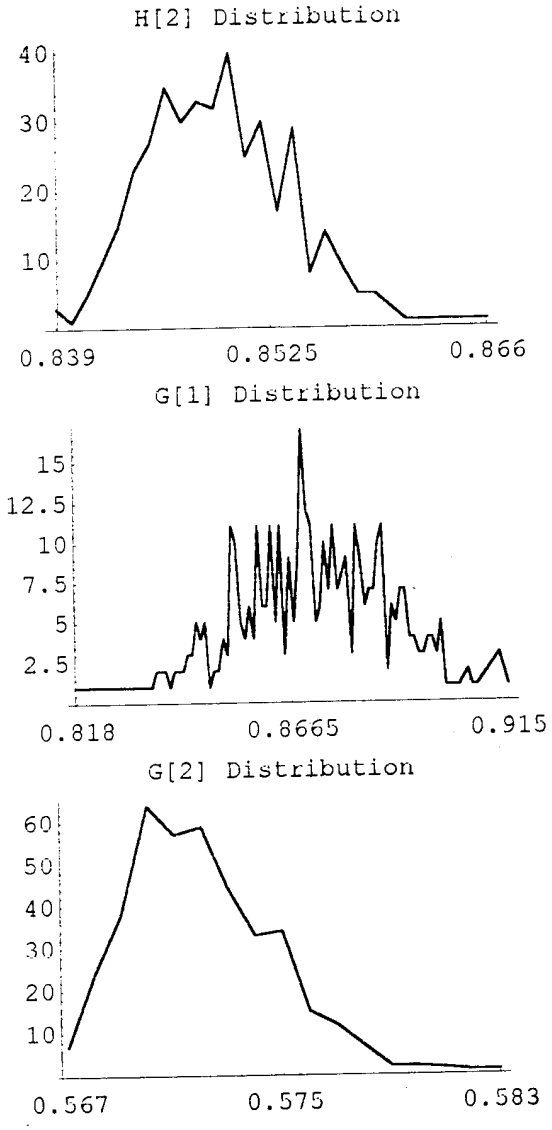
Graphs from independently perturbing each entry in the first row of C by a random draw from a normal distribution with 5% standard deviation.



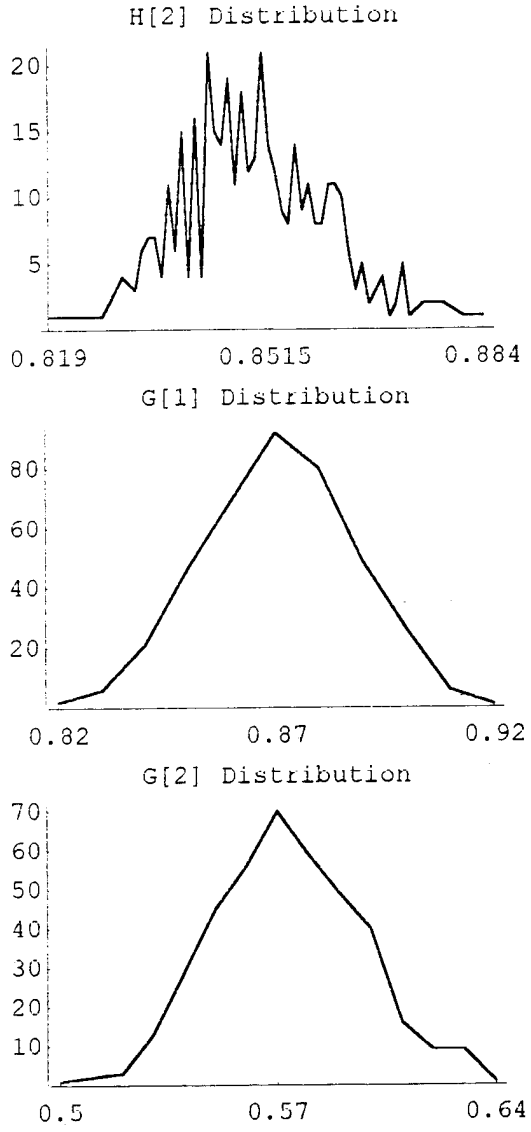




Graphs from simultaneously perturbing each entry in the first column of C by a random draw from a normal distribution with 5% standard deviation.



Graphs from independently perturbing each entry in the first column of C by a random draw from a normal distribution with 5% standard deviation.



This is just a sample of the numerical experiments that we ran. The results led to no substantial conclusions, except perhaps that the ATCAL method is somewhat less sensitive to changes in the initial data. All the methods produced reasonable results (although not consistent results across the methods). Efforts should be made to do large-scale comparisons perhaps using a set of COSAGE replications as a common data pool for all three methods.

§5. Conclusions.

We have examined three methods for computing weapon scores (values, importances). Of the three, the fire allocation methods are the most general, but computationally they are the most difficult. The other two methods, the eigenvalue or potential/antipotential method and the method employed in ATCAL, are both in use in existing combat simulations.

Among the results we have obtained is the proof of the existence and uniqueness for solutions to the ATCAL importance equations, under some mild conditions. This leads to a robust iterative scheme for solving these importance equations. While the ATCAL equations are somewhat ad hoc in nature, they do give reasonable scores for various weapons that are generally consistent with the other methods. (Our numerical experiments were limited to a small number of weapon types on each side (two or three) which limits our conclusions involving comparisons across methods.)

Our most striking observation is that all three methods seem to be based on similar mathematical phenomenon. The potential/antipotential method and the fire allocation methods both result in a kind of "simultaneous eigenvalue problem" – a real simultaneous eigenvalue problem in the first case and a non-linear analogue in the second case. The potential/antipotential method and the method in ATCAL rely on virtually identical proofs for existence and uniqueness, both are centered around a common fixed point theorem. Thus ATCAL yields a different non-linear generalization of the linear eigenvalue method.

The good news is that these similarities give us some assurance, albeit weak assurance, that all three methods will yield generally consistent results. This is confirmed by our relatively limited numerical experiments. Nevertheless the methods do not give identical results and caution is always in order whenever comparisons are being made, because results obtained from one method are not directly comparable to results obtained from another.

The bad news is that we can also expect the methods to suffer from similar qualitative defects and degeneracies. One will probably be able to create examples where all the methods fail to give "reasonable" answers, if you can create an example for any one of the methods. More specifically, many of the criticisms leveled at the eigenvalue approach, in

all likelihood, apply (with appropriate modification) to the other two methods.

The author would again like to express his thanks to the U.S. Army Concepts Analysis Agency and the Army Research Office for their support of this research.

Comprehensive Bibliography

Books and Journal Articles

1) Abraham, Marsden, and Ratiu, "Manifolds, Tensor Analysis and Applications," 2nd ed., Springer-Verlag (1988).

2) Allen, T., "Wargames," McGraw Hill (1987).

3) Arnold, "Stochastic Differential Equations," Wiley Interscience (1974).

4) Barfoot, "The Lanchester Attrition-Rate Coefficient: Some Comments on Seth Bonder's Paper and a Suggested Alternate Method," Operations Research 17 (1969).

5) Bellman, "Dynamic Programming," Princeton University Press (1957).

6) Belopolskaya and Dalecky, "Stochastic Equations and Differential Geometry," Kluwer Academic Publishers (1990).

7) Bickel and Doksum, "Mathematical Statistics," Holden Day (1977).

8) Bonder, "The Mean Lanchester Attrition Rate," Operations Research 18 (1970).

9) Brewer, G. and Shubik, M., "The War Game," Harvard University Press (1979).

10) Burke, "Applied Differential Geometry," Cambridge Univ. Press (1987).

11) Canny, "Generalized Characteristic Polynomials," J. Symb. Comp., Vol. 9 (1990).

12) Chatfield, "The analysis of time series," Chapman & Hall (1975).

13) Crampin and Piran, "Applicable Differential Geometry," London Math. Soc., Lecture Notes 589.

14) Do Carmo, "Riemannian Geometry," Birkhäuser (1992).

15) Dupuy, "Can we rely on computer combat simulations," Armed Forces, J. August (1987).

16) Dubrovin, Fromenko, and Novikov, "Modern Geometry – Methods and Applications," Part I, GTM 93, Springer-Verlag, New York (1984).

17) Elele and Cole, "A New Approach for Modeling Small Conflict Area Gain/Loss, Troop Advance/Retreat Velocity, and Forward Line of Own Troops Movement," Proc. U.S. Army Oper. Res. Symp. (1991).

18) Foias, Sell, and Temin, "Inertial manifolds for nonlinear evolutionary equations," Jour. Diff. Eqn., Vol 73, No. 2 (1988).

19) Frazer, "Elementary matrices and some applications to dynamics and differential equations," Cambridge University Press (1938).

20) Gantmacher, "Matrix Theory," Vol. 2, Chelsea Pub. Co. (1989).

21) Gihman and Skorohad, "Stochastic Differential Equations," Springer-Verlag (1972).

22) Graham, R., "Covariant Formulation of Non-Equilibrium Statistical Thermodynamics," Zeitschrift fur Physik B 26, 397-405 (1977).

23) Grubbs and Shuford, "A New Formula of Lanchester Combat Theory," Operations Research 21 (1973).

24) Hayward, "The Measurement of Combat Effectiveness," Operations Research 16 (1968).

25) Helmbold, "Abstract for Rates of Advance in Historical Land Combat Operations," Proc. U.S. Army O.R. Symposium, Vol. III (1991).

26) Hirsch, "Differential Topology," Springer-Verlag (1988).

27) Hollis, "Yes, we can rely on computer combat simulations," Armed Forces J., October (1987).

28) Howes and Thrall, "Theory of ideal linear weights for heterogeneous combat forces," (1973).

29) Ingber, "Mathematical Comparison of Combat Computer Models to Exercise Data" and "Methodology to Extrapolate Killer-Victim Scoreboards to Higher Echelon Models" (to appear).

30) Kendall, "Multivariate Analysis," Charles Griffen and Co. (1975).

31) Kimbleton, "Attrition Rates for Weapons with Markov-Dependent Fire," Operations Research 19 (1971).

32) Koopman, "A Study of the Logical Basis of Combat Simulation," Opns. Res. 18, pp. 855-882 (1970).

33) Krasnoshchekov and Petrov, "Principles for Constructing Models," (1983).

34) Loental, "Using the Vector-in-Commander Model: Experience and Insights," Proceedings U.S. Army Operations Research Symposium, Vol. III (1991).

35) Mehata and Srinvivasan, "Stochastic Processes," McGraw Hill (1978).

36) Nakahara, "Geometry, Topology, and Physics," Adam Hilger (1990).

37) Oksendal, "Stochastic Differential Equations: An Introduction with Applications," Springer-Verlag (1985).

38) Palmore, "Instability of Computer Simulations of Combat Models," Proc. U.S. Army Oper. Research Symp., Vol. III (1991).

39) Pestien and Sudderth, "Continuous-Time Red and Black: How to Control a Diffusion to a Goal," Mathematics of Operations Research Vol. 10, No. 4 (1985).

40) Przemieniecki, "Introduction to Mathematical Methods in Defense Analysis," AIAA Education Series (1990).

41) Ross, "Stochastic Processes," John Wiley & Sons, New York (1983).

42) Rustagi and Latinen, "Moment Estimation in a Markov-Dependent Firing Distribution," Operations Research 18 (1970).

43) Saigal, "On Piecewise Linear Approximations to Smooth Mappings," Mathematics of Operations Research, Vol. 4, No. 2 (1979).

44) Sayre, "Map Maneuvers and Tactical Rides," 3rd ed. Ft. Leavenworth, Kansas, Army Services School Press (1910).

45) Schuss, Z., "Theory and applications of stochastic differential equations," Wiley Series in Probability, Mathematics, and Statistics 80, John Wiley & Sons, New York (1980).

46) Sibirsky, "Introduction to the algebraic theory of invariants of differential equations," Manchester University Press (1982).

47) Stogan, Mecke, and Kendall, "Stochastic Geometry".

48) Taylor, "Lanchester Models of Warfare," Military Applications Section, ORSA, Vol. 1 (c/o Kendall Inc., 1700 N. Moore St., Suite 1800, Arlington, VA 22209).

49) Taylor, "Lanchester Models of Warfare," Vol. 2.

50) Taylor, "A Note on the Solution to Lanchester-Type Equations with Variable Coefficients," Operations Research 19 (1971).

51) Taylor and Brown, "Canonical Methods in the Solution of Variable Coefficient Lanchester-Type Equations of Modern Warfare," OR, Vol. 24, No. 1 (1976).

52) Watson, R.K., "An application of Martingale Methods to Conflict Models," (1976).

53) Wilks, "Mathematical Statistics," John Wiley & Sons, New York (1962).

54) Wu, "Stochastic Differential Equations," Pitman Advanced Publishing Program, Research Notes in Math. 130 (1985).

Technical Reports

55) Anderson, "Generic Formulas for Calculating Kill Rate Matrices," Institute for Defense Analysis, Alexandria, VA (1988).

56) Anderson and Miercort, "Combat: A Computer Program to Investigated Aimed Fire Attrition Equations, Allocations of Fire, and the Calculation of Weapons Scores," Institute for Defense Analysis, Alexandria VA (1989).

57) Anker and Gafarian, "The validity of assumptions underlying current uses of Lanchester attrition rates," TRAC-WSMR-TD-7-88 (March 1988).

58) Bode, Aldrich, "The Aggregation Problem in Models of Theater Level War," BDM Corp., W-77-129-BR (9/77).

59) Bonder and Farrell, "Development of Analytic Models of Battalion Task Force Activities," Report No. SRL 1957 FR 70-1 (Sept. 1970).

60) Cohen, "The Instability of Linear Heterogeneous Lanchester Equations," AMSAA Technical Report No. 501 (1991).

61) Cohen, "Inner Product Performance Criteria for Evaluating Combat Models," AMSAA Technical Report No. 489 (1991).

62) Comptroller General, "DOD Simulations Improved Assessment Procedures Would Increase the Credibility of Results," Report No. GAO/PEMD-88-3, U.S. GAO, Washington, D.C. (1987).

63) Comptroller General, "Models, Data, and War: A critique of the foundations of Defense Analysis," Report No. PAD-80-21, U.S. GAO, Washington, D.C. (1980).

64) Karr, "A Class of Lanchester Attrition Processes," Institute for Defense Analysis, Alexandria VA (1976).

65) Girardini, "The Army's Conventional Munitions Acquisition Process," Rand Corp. (1988).

66) McCoy, D.H., "Attrition Conference Information Paper," Workshop on Attrition Modeling in Large-Scale Simulations, Feb. 1989, Dallas, TX, US Army Research Office and US Army TRADOC Analysis Command.

67) Perla and Lehoczky, "A new approach to the analysis of stochastic Lanchester processes: 1. Time Evolution," Tech. Report No. 135, Dept. of Stat., Carnegie Mellon Univ., AD A045176 (Sept. 1977).

68) "Proceedings from the Architecture for Stochastic Theater-Level Modeling," Naval Postgraduate School, Monterey, CA (1990).

69) Stockfisch, "Models, Data, and War: A critique of the study of conventional forces," Rand Corp. (1975).

CAA Internal Reports and Documentation

70) AFPSYS Operator's and Programmer's Guide.

71) "ATCAL: An Attrition Model Using Calibrated Parameters," CAA Technical Paper (1983).

72) CEM IV documentation: Technical Handbook and User's Description.

73) CEM VI Vol. 1 Technical Description (1987).

74) CAA packet on "Defender's Advantage".

75) Cooper, "A Heuristic Diffusion Theoretic Approach to FEBA Movement and Maneuver".

76) COSAGE documentation: overview packet, Input/Output Guide, and Division Combat Sample Library.

77) "Improved Method for Calculating Attrition in a Theater Model," CAA Technical Paper (1980).

78) "Stochastic Concepts Evaluation Model (STOCEM)," (1991).